MINIMUM-TIME CONTROL OF THE DAMPED HARMONIC OSCILLATOR VIA MULTI-MODE PWM

A Thesis
Presented to
The Academic Faculty

By

Kartik V. Sastry

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in the
School of Electrical and Computer Engineering

Georgia Institute of Technology

December 2020

MINIMUM-TIME CONTROL OF THE DAMPED HARMONIC OSCILLATOR VIA MULTI-MODE PWM

Thesis committee:

Prof. David Taylor, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Prof. Erik Verriest
School of Electrical and Computer Engineering
Georgia Institute of Technology

Prof. Gregory Durgin
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date approved: December 4, 2020

To all my teachers

ACKNOWLEDGMENTS

I was only able to complete this thesis because I received an abundance of guidance and support. Thus, I would like to express my gratitude to those who supported me and continue to support me in my studies, and in my life.

First and foremost, I thank my advisor, Professor David Taylor. Professor Taylor is the best advisor I could ever ask for. Professor Taylor has helped me in every aspect of this work, from formulation to thesis writing. I am very grateful for his constant encouragement, generous support, thoughtful guidance, and for the trust he places in me. I consider myself extremely fortunate to continue on as his PhD student after the submission of this thesis.

I would also like to thank Professor Erik Verriest and Professor Gregory Durgin, who served on my committee. I am also grateful to Dr. Dave Magee from Texas Instruments and my colleague Aravind Balasubramanian for their inputs on this work.

To Appa, my father, I cannot write enough words to express my gratitude to you for always supporting me in every aspect of my life. Without you, I would not be where I am today.

Finally, I thank Texas Instruments, the Georgia Tech School of ECE, and the Georgia Tech Strategic Energy Institute for supporting me financially during my studies.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

Selected symbols from Chapters 2-4 appear below. Appendix A is self-contained.

Symbol(s)	Defined on	Brief Description
a	pg. 6, 8	single parameter of the dimensionless plant model
τ	pg. 6, 8	dimensionless time variable; $\tau = (\beta/\pi)t$
z	pg. 6, 8	output of the dimensionless plant model; $z = (\kappa\pi)y$
Z	pg. 9	target value of $\lvert z\rvert$ for optimal control purposes
$Z_{\max}$	pg. 13	must choose $Z \in [0, Z_{\max}]$; depends only on a
u	pg. 6, 7	input of the dimensionless plant model;
	pg. 12	solution to CP1/CP2; defined by ψ_1, ψ_2, $\tau_{f,1}$, $\tau_{f,2}$
$\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$	pg. 6	coefficient matrices of the dimensionless plant model
x_1, x_2, $\boldsymbol{x}$	pg. 6	state of the dimensionless plant model
α, β, γ	pg. 7	parameters of the plant model with units
t	pg. 7	time variable with units of seconds
y	pg. 7	output of the plant model with units
v	pg. 7	input of the plant model with units;
	pg. 22	output of H-bridge actuator
V	pg. 7, 22	power supply output; actuator limit; $\lvert v\rvert \leq V$
w_1, w_2	pg. 7	state of the plant model with units
κ	pg. 7	scale factor used in plant model transformation
p_1, p_2, $\boldsymbol{p}$	pg. 10	costate of the dimensionless plant model
ψ_1, ψ_2	pg. 12	phase angles characterizing u
ψ_i	pg. 30	shorthand to refer to ψ_1 and ψ_2
$\tau_{f,1}$, $\tau_{f,2}$	pg. 12	time durations characterizing u
$\tau_{f,i}$	pg. 30	shorthand to refer to $\tau_{f,1}$ and $\tau_{f,2}$
n	pg. 11, 13	discretization parameter in Algorithms 1 and 2
ε	pg. 10, 13	tolerance parameter in Algorithms 1 and 2
A_+, A_-, B_+, B_-	pg. 22	H-bridge control signals;
	pg. 27	PWM module logic signal outputs
c	pg. 27	PWM module counter signal
d_1, d_2	pg. 27, 33	PWM timing / mode waveform shaping parameters
T_{pwm}	pg. 27	PWM counter period in seconds
	pg. 31, 33	mode waveform duration in seconds

Symbol(s)	Defined on	Brief Description
$\tau_{1,1}$, N_1, $\tau_{2,1}$	pg. 30	attributes of $u(\tau)$; sum to $\tau_{f,1}$
$\tau_{1,2}$, N_2, $\tau_{2,2}$	pg. 30	attributes of $u(\tau)$; sum to $\tau_{f,2}$
$\tau_{1,i}$	pg. 30	shorthand to refer to $\tau_{1,1}$ and $\tau_{1,2}$
$\tau_{2,i}$	pg. 30	shorthand to refer to $\tau_{2,1}$ and $\tau_{2,2}$
N_i	pg. 30	shorthand to refer to N_1 and N_2
τ_d	pg. 30	delay in $v(\tau)$; introduced for realization
t_d	pg. 36	delay in $v(\beta t/\pi)$; $t_d = (\pi/\beta)\tau_d$
m	pg. 33	mode waveform index; $m = 0, 1, \ldots, 10$
$\nu_m(t \mid d_1, d_2)$	pg. 33	m^{th} mode waveform; T_{pwm} sec. duration
$\boldsymbol{m}$, $\boldsymbol{d}_1$, $\boldsymbol{d}_2$	pg. 34	store the multi-mode PWM decomposition of $v(\tau - \tau_d)$
$\boldsymbol{m}_1$, $\boldsymbol{d}_{1,1}$, $\boldsymbol{d}_{2,1}$	pg. 34	used to build $\boldsymbol{m}$, $\boldsymbol{d}_1$, $\boldsymbol{d}_2$
$\boldsymbol{m}_2$, $\boldsymbol{d}_{1,2}$, $\boldsymbol{d}_{2,2}$	pg. 34	used to build $\boldsymbol{m}$, $\boldsymbol{d}_1$, $\boldsymbol{d}_2$
R, L, C	pg. 43	true circuit element parameter values
$\hat{R}$, $\hat{L}$, $\hat{C}$	pg. 44	estimates of R, L, and C
s	pg. 44	Laplace transform variable
$\mathcal{L}\{\cdot\}$, $\mathcal{L}^{-1}\{\cdot\}$	pg. 44	forward and inverse Laplace transform operators
$\mathcal{H}(s)$	pg. 44	transfer function of data pre-processing filter
λ	pg. 44	bandwidth of data pre-processing filter
$\mathcal{Y}(s)$, $\mathcal{V}(s)$	pg. 44	Laplace transforms of y and v respectively
$t_1, \ldots, t_M$, M	pg. 44	measurement sampling instants, number of samples
y_0, y_1, y_2	pg. 44	filtered versions of y
v_1	pg. 44	filtered version of v
c_1, c_2, c_3	pg. 44	estimated by least-squares; related to $\hat{R}$, $\hat{L}$, $\hat{C}$
$\hat{\alpha}$, $\hat{\beta}$, $\hat{\gamma}$	pg. 45	estimates of α, β, and γ

SUMMARY

This thesis concerns the continuous-time time-optimal control of systems modeled as linear, time-invariant, second-order oscillators. Motivated by applications in haptics and ultrasonics, we apply time-optimal control to solve the problem of inciting a short-duration oscillatory response from a second-order oscillator. We present two formulations of this control problem, and two novel search algorithms which yield their respective solutions. Both problem formulations are compared from a practical engineering standpoint. Next, we present in detail a novel method for realizing the solution to time-optimal control problems for second-order oscillators. We describe how the pulse-width modulation (PWM) hardware found commonly in modern microcontrollers can be strategically operated to realize the required control signals at the output of an H-bridge circuit. This realization method is termed *multi-mode PWM*. The development of the multi-mode PWM scheme enables the application of time-optimal control to physical systems modeled as second-order oscillators. To demonstrate the applicability of this method, we construct a series RLC circuit and incite a short-duration oscillatory behavior in the loop current.

CHAPTER 1
INTRODUCTION

1.1 Problem Description

This thesis concerns the continuous-time time-optimal control of systems modeled as linear, time-invariant, second-order oscillators. In general, the time-optimal control problem is to determine the control which transfers the system state from a known point in the state space, to another region of the state space, in minimum time. The aim of the work in this thesis is to enable the time-optimal control of physical systems modeled as second-order oscillators. Thus, particular emphasis is placed on the use of a microcontroller to compute the solution to time-optimal control problems in software, and to subsequently operate peripheral circuitry to realize the resulting time-optimal control signals.

The main contribution of this work is a novel strategy for realizing time-optimal controls for linear, time-invariant, second-order oscillators. It is well-known that the solution to continuous-time time-optimal control problems for linear, time-invariant systems is a bang-bang control signal. Such a signal takes on only one of two values (say $\pm U$) at any instant in time. More can be said for the special case of second-order oscillators; the bang-bang time-optimal control signal can be expressed as a time-gated periodic signal [1]. In this thesis, these properties of time-optimal control signals for second-order oscillators are leveraged to develop a method by which commercial off-the-shelf pulse-width modulation (PWM) hardware and an H-bridge actuator can be used to realize time-optimal control signals. The control signals are physically realized as voltage signals, thereby enabling the control of voltage-driven physical systems.

This thesis also features novel search algorithms for the numerical solution of time-optimal control problems for second-order oscillators. The advantages of these search algorithms over standard numerical techniques in optimal control are highlighted. These search algorithms are simple and not memory intensive, making them ideal for microcontroller implementation. The search algorithms and control realization scheme developed in this thesis apply equally well to the solution of any time-optimal control problem for second-order oscillators. However, in this thesis, emphasis is placed on time-optimal control problems of physical significance for ultrasonics and haptics applications which were the initial motivations for pursuing this work. Thus, two time-optimal control problems are formulated and solved in this thesis. The goal of Control Problem 1 is to drive the oscillator state from the origin of the state space to a target set in minimum time, and subsequently drive the state back to the origin in minimum time. The goal of Control Problem 2 is to find the minimum-time state trajectory which starts and ends at the origin of the state space but touches a target set at some point in the excursion away from the origin. Both problems are slightly different, but have the same objective of inciting a short-duration oscillatory response from the system. The physical significance of operating ultrasonic and haptic systems in this manner is outlined below.

1.2 Motivations

Piezoelectric ultrasonic transducers convert electrical energy to acoustical energy and vice-versa. These devices are commonly deployed on the rear bumpers of automobiles for obstacle detection. By design, piezoelectric transducers are lightly damped oscillators, and are operated near their resonance frequencies for efficient energy conversion. Near resonance, these systems are well modeled by third-order dynamics (using the Butterworth Van-Dyke model). Modeling equations are presented in Appendix A, and it is shown that a second order model is adequate in the presence of

time-scale separation between the (fast) electrical dynamics and the (slow) mechanical dynamics. Exciting the electrical terminals of a piezoelectric ultrasonic transducer incites the transmission of a pressure wave from the acoustic port of the transducer.

Using piezoelectric ultrasonic transducers for short-range distance sensing is a task of interest to the ultrasonics community. In order to perform short-range sensing with a single transducer and a pulse-echo operating strategy, one must command the device to emit a short-duration pressure wave. Furthermore, this pressure wave must have sufficient amplitude to overcome attenuation along its journey towards a reflector and back to the transducer. Shorter-duration pressure waves result in a shorter minimum sensing distance for the device; thus the use of time-optimal control for this task is natural. It is shown in this thesis that Control Problem 1 and Control Problem 2 solve this problem when the target set is defined appropriately.

Linear resonant actuators convert electrical energy to mechanical energy and vice-versa. These devices are commonly embedded in mobile phones in order to generate haptic feedback for a human user. By design, linear resonant actuators are lightly damped oscillators. These systems are also well modeled by third-order dynamics. Modeling equations are presented in Appendix A, and it is shown that a second order model is adequate in the presence of time-scale separation between the (fast) electrical dynamics and the (slow) mechanical dynamics. Exciting the electrical terminals of a linear resonant actuator incites mechanical vibration of the device and any object in contact with the device (such as a mobile phone).

Using linear resonant actuators for the generation of short-duration, crisp haptic effects is a task of interest to the haptics community. In order to generate short-duration haptic effects like the "haptic click" and the "haptic scroll-wheel" effects, one must command the device to vibrate for a short interval of time. Furthermore, this vibration must have sufficient amplitude in order to be perceived by a human being. Shorter-duration vibrations result in more 'crisp' haptic effects; thus the use

of time-optimal control for this task is natural. It is shown in this thesis that Control Problem 1 and Control Problem 2 solve this problem when the target set is defined appropriately.

1.3 Survey of Existing Work and Differentiating Aspects

The setpoint tracking problem for second-order oscillators is similar in spirit to the problem considered in this thesis. A fundamental distinction between the problem in this thesis and that of setpoint tracking is that there is no need to hold the oscillator state at the target for our purposes. Nonetheless, the target state must be reached as rapidly as possible. Much work has been devoted to driving oscillatory systems to a target state while minimizing residual vibration in tracking. Simple feedback control methods called Integral Resonant Control (IRC) and Positive Position Feedback (PPF) are implemented in [2, 3, 4]. These controllers introduce additional damping with the goal of reducing oscillations in tracking tasks. These controllers are effective and simple, but do not explicitly account for actuator saturation in their design. Hence, no guarantees can be made on tracking time once actuator saturation is enforced. Furthermore, implementation requires continuous variation of control amplitude, which complicates actuator design.

Input shaping is another common setpoint tracking technique, which initially appears to solve the problem considered in this thesis. Here, an FIR filter is introduced between a reference command generator and the oscillatory plant [5]. The filter decimates any component of the reference signal which excites oscillations in the plant, allowing for setpoint tracking with minimal residual vibration. Input shapers have been designed to be robust to modeling errors [6]. Furthermore, input shaping allows for actuator saturation to be explicitly accounted for, and also allows for minimization of tracking time, subject to actuator saturation and residual vibration constraints [7]. While input shaping is effective, implementation once again requires continuous vari-

ation of control amplitude. It was also shown in [8] that the performance of input shapers degrades with significant actuator quantization.

Regulation of the damped oscillator to the origin has also been studied well. Buhl and Lohmann explicitly account for actuator saturation in [9], and present a controller design based on a single parameter for decay-rate. A variable decay-rate controller is also presented which approaches the time-optimal result. While elegant, this control scheme still requires that the control input have a continually variable amplitude. In [10], Braker and Pao propose near time-optimal controllers for regulation from initial conditions to the origin, as well as setpoint tracking. As in [9], this solution has the drawback of requiring that the control input have a continually variable amplitude.

Texas Instruments' commercially available DRV2603 Linear Resonant Actuator Driver also performs feedback control to bring a linear resonant actuator to rest, following a period of excitation [11, 12]. The complete details of the control scheme employed by this product are not publicly available. However, system response plots indicate that a feedback controller operating on the principle of pulse-amplitude-modulation is incorporated in the device. This controller appears to approach the time-optimal result. However, if the plant in concern is truly a linear, time-invariant, stable, second-order oscillator, the theory of time-optimal control states that the time-optimal control law is bang-bang. In contrast to the solution proposed in this thesis, the DRV2603 also requires a variable control amplitude.

CHAPTER 2

CONTROL PROBLEM FORMULATIONS AND SOLUTIONS

This work concerns the time-optimal control of linear, time-invariant, stable, second-order oscillators. There are many physical systems that are well modeled as second order oscillators, and many types of boundary conditions that can be used to define time-optimal control problems. It is important to note that the methods in Chapter 2 and Chapter 3 apply generally, even though they are presented in the context of a particular time-optimal control problem of interest.

2.1 A Universal Plant Model for Second Order Oscillators

The time-optimal control problem formulations in this thesis will use a fully dimensionless, single-parameter state-space model for the second-order oscillator, given by:

$$\frac{d}{d\tau}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \underbrace{\begin{bmatrix} -a & \pi \\ -\pi & -a \end{bmatrix}}_{\boldsymbol{A}} \underbrace{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}}_{\boldsymbol{x}} + \underbrace{\begin{bmatrix} 0 \\ a \end{bmatrix}}_{\boldsymbol{B}} u \tag{2.1a}$$

$$z = \underbrace{\begin{bmatrix} -a & \pi \end{bmatrix}}_{\boldsymbol{C}} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{2.1b}$$

In this model, state variables x_1 and x_2, control input u, output signal z, time τ, and the single parameter a are all dimensionless. The single parameter a captures the level of damping in the oscillator, and is positive for internally stable oscillators. Equations to follow will demonstrate how the dimensionless model in Equation 2.1 may be derived from a more common state-space model which involves variables with

units.

To obtain meaningful solutions to time-optimal control problems, the control input is assumed to be bounded in amplitude. When solving time-optimal control problems involving the dimensionless plant model, we require $|u(\tau)| \leq 1$. Since the time-optimal control signal is known to be bang-bang, it is worth noting that at any time instant τ, $u(\tau) \in \{\pm 1, 0\}$. An important feature in the following set of equations is the relationship between the input of the dimensionless plant model (u) and the input to a physical plant (v) which is provided by an actuator with output limits.

A state-space model in the form of Equation 2.2 is commonly obtained by applying the laws of physics:

$$\frac{d}{dt}\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -(\alpha^2+\beta^2) & -2\alpha \end{bmatrix}\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \gamma \end{bmatrix} v \tag{2.2a}$$

$$y = \begin{bmatrix} 0 & 1 \end{bmatrix}\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \tag{2.2b}$$

Consequently, state variables w_1 and w_2, control input v, output signal y, time t, and model parameters α, β, and γ will have physical interpretations and units. For example, in a voltage-driven series RLC circuit, v is the source voltage, w_1 is the charge on the capacitor and $y = w_2$ is the loop current. Model parameters α, β and γ depend only on circuit component values R, L and C. More examples are presented in Appendix A.

Suppose that the control signal in Equation 2.2a is constrained by $|v(t)| \leq V$ due to actuator limits. We can transform Equation 2.2 into Equation 2.1 by defining:

$$\kappa := \frac{\alpha}{\gamma V} \quad , \quad u := \frac{v}{V} \quad , \quad z := (\kappa\pi)\, y \tag{2.3}$$

and making the following variable changes:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} := \kappa \begin{bmatrix} \beta & 0 \\ \alpha & 1 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \quad , \quad \tau := \left(\frac{\beta}{\pi}\right) t \quad , \quad a := \frac{\pi\alpha}{\beta} \tag{2.4}$$

The transformation from Equation 2.2 to Equation 2.1 reduces the number of model parameters from three (α, β, γ) to one (a). Furthermore, the effects of units are completely removed - both state variables take values on the same order of magnitude, and time scaling removes the dependence on the true frequency of oscillation of the plant; the plant of Equation 2.1 completes one cycle of oscillation in 2 units of scaled time τ. These properties of the plant model in Equation 2.1 are advantageous when performing numerical computations. For example, the single-parameter model can be used for faithful computer simulation of system dynamics at any scale (in time and amplitude) without concern for issues related to finite-precision arithmetic. When applying time-optimal control to physical systems, a model of the form of Equation 2.2 will be obtained first, and subsequently transformed into a model of the form of Equation 2.1.

Time optimal control problems also require the definition of a *target set* - a region of the state space to be reached in minimum time, starting from a known point in the state space. Defining a target set in the context of the applications of interest requires committing to a definition of the "intensity" of the short-duration oscillatory response that we seek to incite from the oscillator. In this thesis, we use output equations to define measures of oscillation "intensity". There is no universally accepted metric for this problem. We therefore *arbitrarily* select the second state variable of Equation 2.2 as our "intensity" metric, which gives rise to Equation 2.2b and Equation 2.1b. However, other definitions of "intensity" could be used as well. For example, in a series RLC circuit, defining "intensity" in terms of the loop current or the voltage/charge on the capacitor would lead to Equation 2.2b being linear in w_1 and w_2. However,

"intensity" could also be defined in terms of the total energy stored in the capacitor/inductor ensemble, in which case Equation 2.2b would be a quadratic function of w_1 and w_2. When applying time-optimal control to physical systems, an "intensity" metric will be specified in terms of physical state variables by defining Equation 2.2b. Equation 2.2b will subsequently be transformed into Equation 2.1b.

2.2 Control Problem 1 (CP1)

2.2.1 Formulation

The goal of Control Problem 1 (CP1) is to drive the oscillator state from the origin of the state space to a target set in minimum time, and subsequently drive the state back to the origin in minimum time. Performing this *rest-to-rest maneuver* will produce a waveform $z(\tau)$ with a narrow, sharply peaked envelope. One way to meet this objective is to *solve two time-optimal control problems in sequence*, where both problems are an instance of a more general problem in which the oscillator described by Equation 2.1 is driven from an initial state $\boldsymbol{x}_0$ to a target set $\mathcal{S}$ in the state space in minimum time:

$$\begin{aligned} &\text{minimize } \int_0^{\tau_f} 1 \; d\tau \\ &\text{subject to : } \frac{d}{d\tau}\boldsymbol{x} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}u, \;\; |u(\tau)| \leq 1 \\ &\qquad\qquad z = \boldsymbol{C}\boldsymbol{x} \\ &\qquad\qquad \boldsymbol{x}(0) = \boldsymbol{x}_0, \;\; \boldsymbol{x}(\tau_f) \in \mathcal{S} \end{aligned} \tag{2.5}$$

To solve CP1, we must solve the following two subproblems in sequence:

1. *Subproblem 1a (Startup):* Drive the oscillator state from rest ($\boldsymbol{x} = \boldsymbol{0}$) to a *target output level*, encoded by $|z| \geq Z$. For this problem, define $\boldsymbol{x}_0 := \boldsymbol{0}$ and $\mathcal{S} := \{\boldsymbol{x} \in \mathbb{R}^2 \;:\; |z| \geq Z\}$.

2. *Subproblem 1b (Shutdown):* Drive the oscillator state from some known initial state $\boldsymbol{x}_0$ to the origin of the state space. Ideally, one would define $\mathcal{S} := \{0\}$. However, when solving this subproblem numerically, it becomes necessary to introduce a small parameter, ε, and relax the definition of the target set to $\mathcal{S} := \{\boldsymbol{x} \in \mathbb{R}^2 \ : \ \|\boldsymbol{x}\|_2 \leq \varepsilon\}$ for reasons that will be made clear shortly.

In order to achieve the desired shaping of z, we first solve *Subproblem 1a*, record the terminal state, and subsequently solve *Subproblem 1b* using the terminal state of *Subproblem 1a* as the initial condition. Exploiting time-invariance, the solution to CP1 can be formed by appropriately shifting and summing the solutions to *Subproblem 1a* and *Subproblem 1b*.

2.2.2 Solution via Two, One-Dimensional Searches

Application of the necessary conditions for optimality to the general problem in Equation 2.5 yields the costate dynamics $\frac{d}{d\tau}\boldsymbol{p} = -\boldsymbol{A}'\boldsymbol{p}$, where $\boldsymbol{p}$ is the costate vector with elements p_1 and p_2 [1]. It follows that:

$$\begin{bmatrix} p_1(\tau) \\ p_2(\tau) \end{bmatrix} = e^{a\tau} \begin{bmatrix} \cos(\pi\tau) & \sin(\pi\tau) \\ -\sin(\pi\tau) & \cos(\pi\tau) \end{bmatrix} \begin{bmatrix} p_1(0) \\ p_2(0) \end{bmatrix}$$

Using the phasor addition theorem, we may write:

$$p_2(\tau) \propto e^{a\tau} \sin(\pi\tau + \psi), \quad \psi \in [0, 2\pi) \tag{2.6}$$

where the phase angle ψ is a function of $p_1(0)$ and $p_2(0)$. Application of Pontryagin's minimum principle to this problem yields the bang-bang solution $u(\tau) = -\text{sign}(p_2(\tau))$. Using Equation 2.6, we deduce that the time-optimal control which solves the general

problem in Equation 2.5 is of the form:

$$u(\tau) = \text{sign}(\sin(\pi\tau + \psi)),\ \psi \in [0, 2\pi) \tag{2.7}$$

Determining $u(\tau)$ amounts to determining the unknown initial costates $p_1(0)$ and $p_2(0)$, or equivalently, ψ. In general, a two-point boundary value problem must be solved to determine $u(\tau)$ by way of $p_1(0)$ and $p_2(0)$, or a two dimensional search for $(p_1(0), p_2(0))$ must be performed over an unbounded plane. Reformulating the problem in terms of phase angle ψ allows for the determination of ψ and associated minimum time τ_f (and therefore $u(\tau)$) via a simple search procedure shown in Algorithm 1. Three parameters (including discretization parameter n, which appears explicitly) must be set before executing Algorithm 1. These parameters will be discussed in the final subsection of this chapter.

Algorithm 1 Search-Based Solution to Subproblem 1a/1b

$\tau_{\text{array}}, \phi_{\text{array}} \leftarrow$ empty $n \times 1$ array
for $i = 0, 1, \ldots, n-1$ **do**
  $\phi_{\text{array}}[i] \leftarrow \frac{2\pi i}{n-1}$
  $u(\tau) \leftarrow \text{sign}(\sin(\pi\tau + \phi_{\text{array}}[i]))$
  Solve IVP in Equation 2.1 with $\boldsymbol{x}(0) = \boldsymbol{x}_0$ until $\boldsymbol{x}(\tau) \in \mathcal{S}$
  $\tau_{\text{array}}[i] \leftarrow \tau$
end for
$i_{\min} \leftarrow \text{argmin}\{\tau_{\text{array}}\}$
$\psi \leftarrow \phi_{\text{array}}[i_{\min}]$
$\tau_f \leftarrow \tau_{\text{array}}[i_{\min}]$

Upon termination of Algorithm 1, the solution to the problem in Equation 2.5 will be completely characterized by the pair (ψ, τ_f). Since Algorithm 1 must be executed twice to solve CP1, the final solution will be characterized by two pairs: $(\psi_1, \tau_{f,1})$ and

$(\psi_2, \tau_{f,2})$. More concretely, the solution to CP1 is of the form:

$$u(\tau) = \begin{cases} \text{sign}(\sin(\pi\tau + \psi_1)) & , \tau \in [0, \tau_{f,1}) \\ \text{sign}(\sin(\pi(\tau - \tau_{f,1}) + \psi_2)) & , \tau \in [\tau_{f,1}, \tau_{f,1} + \tau_{f,2}) \\ 0 & , \text{otherwise} \end{cases} \tag{2.8}$$

2.3 Control Problem 2 (CP2)

2.3.1 Formulation

The goal of Control Problem 2 (CP2) is to find the minimum-time state trajectory which starts and ends at the origin of the state space, but touches a target set at some point in the excursion away from the origin. We formulate this problem as:

$$\begin{aligned} \text{minimize } & \int_0^{\tau_{f,1}+\tau_{f,2}} 1 \; d\tau \\ \text{subject to: } & \frac{d}{d\tau}\boldsymbol{x} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}u, \;\; |u(\tau)| \leq 1 \\ & z = \boldsymbol{C}\boldsymbol{x} \\ & \boldsymbol{x}(0) = \boldsymbol{0}, \;\; |z(\tau_{f,1})| \geq Z, \;\; \|\boldsymbol{x}(\tau_{f,1} + \tau_{f,2})\|_2 \leq \varepsilon \end{aligned} \tag{2.9}$$

Note that the target set $\mathcal{S} := \{\boldsymbol{x} \in \mathbb{R}^2 \;:\; |z| \geq Z\}$ must be reached at time $\tau = \tau_{f,1}$, and that the state vector must lie inside a circle of radius ε (centered at the origin of the state space) at time $\tau = \tau_{f,1} + \tau_{f,2}$. The need for small parameter ε is described in the final subsection of this chapter.

2.3.2 Solution via One, Two-Dimensional Search

The solution to CP2 (Equation 2.9) is also of the form of Equation 2.8. To solve CP2, we determine the pair (ψ_1, ψ_2) which minimizes the time taken to complete the rest-to-rest maneuver, i.e. the *sum* $\tau_{f,1} + \tau_{f,2}$. The solution is readily obtained

by the search procedure in Algorithm 2. Three parameters (including discretization parameter n and tolerance paramter ε, which appear explicitly) must be set before executing Algorithm 2. These parameters will be discussed in the next subsection.

Algorithm 2 Search-Based Solution to Control Problem 2

$\tau_{\text{array}} \leftarrow$ empty $n \times n$ array
$\phi_{1,\text{array}}, \tau_{\text{up, array}} \leftarrow$ empty $n \times 1$ array
for $i = 0, 1, \ldots, n-1$ **do**
  $\phi_{1,\text{array}}[i] \leftarrow \frac{2\pi i}{n-1}$
  $u(\tau) \leftarrow \text{sign}(\sin(\pi\tau + \phi_{1,\text{array}}[i]))$
  Solve IVP in Equation 2.1 with $\boldsymbol{x}(0) = \mathbf{0}$ until $|z(\tau)| \geq Z$
  Record terminal state as $\boldsymbol{x}_0$
  $\tau_{\text{up, array}} \leftarrow \tau$
  $\phi_{2,\text{array}} \leftarrow$ empty $n \times 1$ array
  for $j = 0, 1, \ldots, n-1$ **do**
    $\phi_{2,\text{array}}[j] \leftarrow \frac{2\pi j}{n-1}$
    $u(\tau) \leftarrow \text{sign}(\sin(\pi\tau + \phi_{2,\text{array}}[j]))$
    Solve IVP in Equation 2.1 with $\boldsymbol{x}(0) = \boldsymbol{x}_0$ until $\|\boldsymbol{x}(\tau)\|_2 \leq \varepsilon$
    $\tau_{\text{array}}[i, j] \leftarrow \tau_{\text{up, array}}[i] + \tau$
  end for
end for
$(i_{\min}, j_{\min}) \leftarrow \text{argmin}\{\tau_{\text{array}}\}$
$\psi_1 \leftarrow \phi_{1,\text{array}}[i_{\min}]$
$\psi_2 \leftarrow \phi_{2,\text{array}}[i_{\min}]$
$\tau_{f,1} \leftarrow \tau_{\text{up, array}}[i_{\min}]$
$\tau_{f,2} \leftarrow \tau_{\text{array}}[i_{\min}, j_{\min}] - \tau_{f,1}$

Upon termination of Algorithm 2, the solution to CP2 will be completely characterized by $(\psi_1, \psi_2, \tau_{f,1}, \tau_{f,2})$.

2.4 Comparison of Formulations and Solutions

Solving the optimal control problems of interest via the search algorithms presented is preferable to employing standard solution methods. It is well known from the theory of optimal control that in general, a two-point boundary value problem must be solved to obtain the time-optimal control [1]. Numerical boundary value problem solvers rely on iterative, non-parallelizable algorithms that require the user

to set several convergence-related parameters, which is an arduous and ideally avoidable task. Furthermore, these algorithms often employ some form of gradient descent, which makes their performance dependent on the quality of a user-provided initial guess, and the existence of local extrema (or lack thereof). If the user-provided convergence parameters and initial guess are not set appropriately, these algorithms may converge to a local extremum, or fail to converge to a solution at all. In contrast, Algorithm 1 and Algorithm 2 converge to the global extremum, require no initial guess, and require the user to set three parameters which are straightforward to determine. Additionally, each iteration of Algorithm 1 and Algorithm 2 is independent, so both algorithms are readily parallelizable if computation time is of concern and parallel computing resources are available. While it is true that there are parameter choices that can cause Algorithm 1 or Algorithm 2 to not converge to a solution, it is straightforward to determine how to modify the parameter selection to ensure convergence to an acceptable solution. In contrast, modifying the many input parameters of a numerical boundary value problem solver to ensure convergence is a challenging task.

As with any optimal control solver, Algorithm 1 and Algorithm 2 can only produce a solution if a solution exists. It is therefore necessary to ensure that the optimal control problem to be solved is *feasible* before attempting to execute Algorithm 1 or Algorithm 2. The plant model of Equation 2.1 is completely specified by a user-provided value of parameter a. The user must also provide a parameter Z, which specifies a target set to be reached. For a fixed a, one could consider many values of Z. To ensure problem feasibility, one must choose $Z \in [0, Z_{\max}]$ (for a stable plant), where $Z_{\max}$ depends only on a, and is equal to the maximum value of $|z(\tau)|$ achieved in periodic steady state if $u(\tau) = \text{sign}(\sin(\pi\tau))$ was applied in perpetuity. $Z_{\max}$ can be thought of as a characteristic of the plant.

After specifying plant parameter a and a feasible value of Z, the following three parameters must be set prior to executing Algorithm 1 or Algorithm 2:

The three parameters that appear in Algorithm 1 and Algorithm 2 are:

- *Discretization parameter: n* - This parameter controls the size of the phase angle search space. This space is bounded for both Algorithm 1 and Algorithm 2. This search space is one-dimensional in Algorithm 1, and two-dimensional in Algorithm 2. *Parameter n should be set as large as possible.*

- *Discretization parameter: time step* - This parameter is hidden within the "Solve IVP in Equation 2.1..." statements appearing in Algorithm 1 and Algorithm 2. A variety of techniques exist for solving initial value problems (IVP's) via numerical integration. For simplicity, it is assumed that a fixed step size method (such as the Forward Euler method) is used, so that only a single parameter (the fixed size time-step) needs to be specified. However, this is not a limitation of Algorithm 1 or Algorithm 2; more exotic variable step-size methods can be used to solve IVP's if desired. One advantage of using the single-parameter, dimensionless model of Equation 2.1 when solving IVP's is that the fixed step size can be chosen without considering the characteristic time scales of the physical system under consideration. *The time step parameter should be set as small as possible.*

- *Tolerance parameter: ε* - This parameter is needed to ensure that Algorithm 1 and Algorithm 2 converge to a solution. Discretization of time and the phase angle search space introduces small errors into the solution obtained by the search algorithms. To understand the nature of these errors, consider solving Subproblem 1b via Algorithm 1. In this problem, we seek a phase angle ψ_2 and a time $\tau_{f,2}$ such that application of $u(\tau) = \text{sign}(\sin(\pi\tau + \psi_2))$ for $\tau_{f,2}$ units of (scaled) time drives the oscillator from $\boldsymbol{x}(0) = \boldsymbol{x}_0$ to $\boldsymbol{x}(\tau_{f,2}) = 0$. Suppose that the true solutions, ψ_2^* and $\tau_{f,2}^*$, are somehow known.

  - *Error due to time discretization:* $\boldsymbol{x}(\tau_{f,2}^*)$ is computed by numerically in-

tegrating Equation 2.1a using $u(\tau) = \text{sign}(\sin(\pi\tau + \psi_2^*))$. Due to the approximations made in numerical integration (i.e. approximating an integral by a Riemann sum), the computed vector $\boldsymbol{x}(\tau_{f,2}^*)$ will not exactly be the zero vector, but will approach the zero vector as the time step used for numerical integration decreases.

– *Error due to phase angle discretization:* The phase angle search space is discretized into n discrete values between 0 and 2π. Suppose that the exact solution, ψ_2^*, is 'skipped over' due to discretization, and that $\hat{\psi}_2$ is the closest phase angle to ψ_2^* in the discretized search space. Suppose that we compute $\boldsymbol{x}(\tau_{f,2}^*)$ by *exactly* integrating Equation 2.1a using $u(\tau) = \text{sign}(\sin(\pi\tau + \hat{\psi}_2))$. The computed vector $\boldsymbol{x}(\tau_{f,2}^*)$ will not exactly be the zero vector, but will approach the zero vector as n increases (such that $\hat{\psi}_2$ is closer to ψ_2^*).

It is expected that the true solution can be approximated with great fidelity by setting n to be large and the time step to be small. Nonetheless, it is unlikely that the condition $\boldsymbol{x} = \boldsymbol{0}$ will be met *exactly* during the search, due to small approximation errors. Thus, when driving the oscillator state from the target set back to the origin of the state space, relaxing the definition of the target set from $\mathcal{S} = \{0\}$ to $\mathcal{S} = \{\boldsymbol{x} \in \mathbb{R}^2 \ : \ \|\boldsymbol{x}\|_2 \leq \varepsilon\}$ ensures that Algorithm 1 or Algorithm 2 finds a solution. *Parameter ε should be set as small as possible, bearing in mind that the choices of n and the time step parameter will effectively lower-bound the value of ε that can be chosen.*

CP1 and CP2 were presented as competing solutions to the overall goal of inciting a short-duration oscillatory response from a second-order oscillator. Solving CP1 required the solution of two time-optimal control problems, both of which were solved by Algorithm 1. Solving CP2 required the solution of a single time-optimal control

problem via Algorithm 2. The remainder of this subsection is devoted to a comparison between these two competing solution methods.

In the notation of CP2, CP1 can be thought of as seeking to independently minimize $\tau_{f,1}$ (time taken to reach the target set) and $\tau_{f,2}$ (time to return to the origin). In contrast, CP2 seeks to minimize the total rest-to-rest time - the sum $\tau_{f,1} + \tau_{f,2}$. It follows from the triangle inequality that the rest-to-rest time obtained by solving CP2 will always be less than or equal to the rest-to-rest time obtained by CP1. This fact is illustrated by means of a numerical example in Figure 2.1.

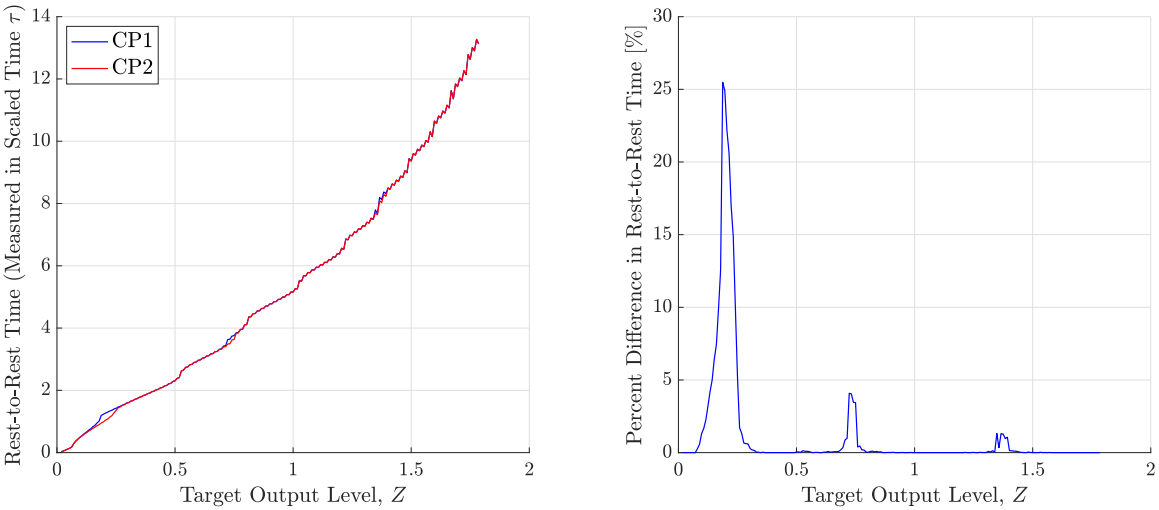


Figure 2.1: Rest-to-rest times obtained by solving CP1 and CP2

In simulation, CP1 and CP2 were solved for an oscillator characterized by $a = 0.2001$. Parameter values of $n = 400$, $\varepsilon = 10^{-4}$, were selected prior to executing Algorithm 1 and Algorithm 2. IVP's were solved numerically using the Forward Euler method and a fixed time step of 1/2000. The output signal was defined according to Equation 2.1b. 100 values of target output Z were chosen between zero and the (maximum) amplitude of $z(\tau)$ achieved in steady state (if the stable plant were to be excited by a square wave at the resonant frequency in perpetuity). As an aside, the particular choice of a used for this short study corresponds to the series RLC circuit used for validation in Chapter 4, and the choice of z corresponds to measuring

oscillation "intensity" in terms of the loop current.

The rest-to-rest times produced by either method are plotted as a function of target output Z in the left panel of Figure 2.1. This plot indicates that across all possible values of Z, for the particular oscillator under consideration, the two solution methods produce rest-to-rest times that are nearly identical. The percent degradation in rest-to-rest time obtained by choosing to solve CP1 over CP2 is plotted in the right panel of Figure 2.1.

As expected, the rest-to-rest time obtained by solving CP2 is always less than or equal to that of CP1. This benefit comes at the cost of increased computation - solving CP2 requires the use of Algorithm 2, which has $\mathcal{O}(n^2)$ time complexity, whereas solving CP1 via Algorithm 1 (two times), requires just $\mathcal{O}(n)$ computation time.

The plots in Figure 2.1 indicate that for $Z \geq 0.4$, CP1 and CP2 yield rest-to-rest times that differ by less than 5% (often much less than 5%). The worst-case discrepancy in rest-to-rest time of about 25% is observed around $Z = 0.2$. However, referring to the left panel of Figure 2.1, we note that the actual rest-to-rest times for $Z < 0.4$ are less than 2 units of scaled time τ (or equivalently, less than $\frac{2\pi}{\beta}$ seconds). Inciting responses from oscillatory systems that are less than 2 units of scaled time τ in duration is not of practical interest. In fact, the system response would not even exhibit oscillatory behavior in this case, since the total duration of the response is less than one cycle of oscillation. By this logic, the rest-to-rest times obtained by solving CP1 and CP2 are practically equivalent, for practically relevant settings of target output Z.

Since the discretization of phase angle represents an approximation to the true solution, it is desirable to increase n as much as possible, which creates a trade-off with computation time. Therefore, from a practical engineering perspective, it is preferable to adopt CP1 as the solution method of choice. In choosing to solve CP1

over CP2, we save computation time, with practically no required sacrifice in rest-to-rest time. Viewed another way, for a fixed, desired computation time, solving CP1 allows for a denser search over phase angle than if CP2 were to be solved, thereby producing a closer approximation to the true, continuous-time solution.

CHAPTER 3
REALIZATION VIA MULTI-MODE PWM

In this thesis, *realization* refers to the production of a voltage waveform which resembles a known function of time. This chapter introduces *multi-mode PWM*, a novel method for realizing bang-bang control waveforms which are solutions to time-optimal control problems for second-order oscillators. The need for a realization method arises when one seeks to apply time-optimal control to voltage-driven physical systems which are modeled as second-order oscillators.

In this chapter, we assume that a voltage-driven physical system is initially modeled by Equation 2.2, and the constant coefficients α, β, and γ of this model have been obtained through parameter identification. We also assume knowledge of the actuator limit, V. We assume that CP1 has been solved after transforming the initial plant model into the dimensionless model of Equation 2.1. We assume that the resulting dimensionless time-optimal control signal $u(\tau)$ is known by way of the four constants, ψ_1, $\tau_{f,1}$, ψ_2, and $\tau_{f,2}$, which parameterize Equation 2.8. To enable experimentation, we must use this information to produce the signal $v = Vu$ which drives the physical system. *Multi-mode PWM leverages the capabilities of a typical microcontroller to produce v as a voltage waveform at the output of an H-bridge actuator.*

This chapter is organized as follows. The bang-bang nature of control signals u and v necessitates the use of an H-bridge actuator to drive the plant. Therefore, operation of the H-bridge is discussed in Section 3.1. Next, a naive microcontroller-based realization strategy and its drawbacks are presented in Section 3.2 to motivate the more sophisticated strategy called multi-mode PWM. Section 3.3 then describes the operation of a PWM module, a piece of programmable hardware found in modern microcontrollers, that will be used to command the voltage output of the H-bridge

actuator. In particular, the interplay between the PWM module's programmable configuration parameters is highlighted in Section 3.3. With the operation of a PWM module in mind, the *multi-mode PWM decomposition* is presented in Section 3.4. The multi-mode PWM decomposition maps the time-optimal control signal u into sequences of PWM module configuration parameters, to be issued to a PWM module over time. It is shown that operating a PWM module in this manner leads to the production of the desired plant input signal v at the output of an H-bridge controlled by the PWM module. Finally, additional details related to a microcontroller implementation of multi-mode PWM are provided in Section 3.5.

3.1 Operation of an H-bridge

At any time instant, the plant input signal v takes on a value in $\{\pm V, 0\}$. Therefore, the actuator used to realize v must be capable of producing a bipolar voltage at its output, on an instantaneous basis. The ideal H-bridge shown in Figure 3.1 is well-suited for this task.

An ideal voltage source supplies a constant $+V$ volts to one side of the ideal H-bridge. The four transistors are idealized as electrically-controlled switches which can be opened and closed by controlling the four logic signals A_+, A_-, B_+, and B_-. A microcontroller manipulates the four logic signals over time to achieve a desired variation in output v. Although it initially appears that there are $2^4 = 16$ possible configurations of the four logic signals, safe operation of the device requires that power and ground never be shorted together. Thus, the constraints

$$A_- := \overline{A_+} \quad \text{and} \quad B_- := \overline{B_+} \tag{3.1}$$

must be enforced at every time instant (the overbar denotes logical negation). The constraints of Equation 3.1 reduce the number of valid H-bridge configurations to

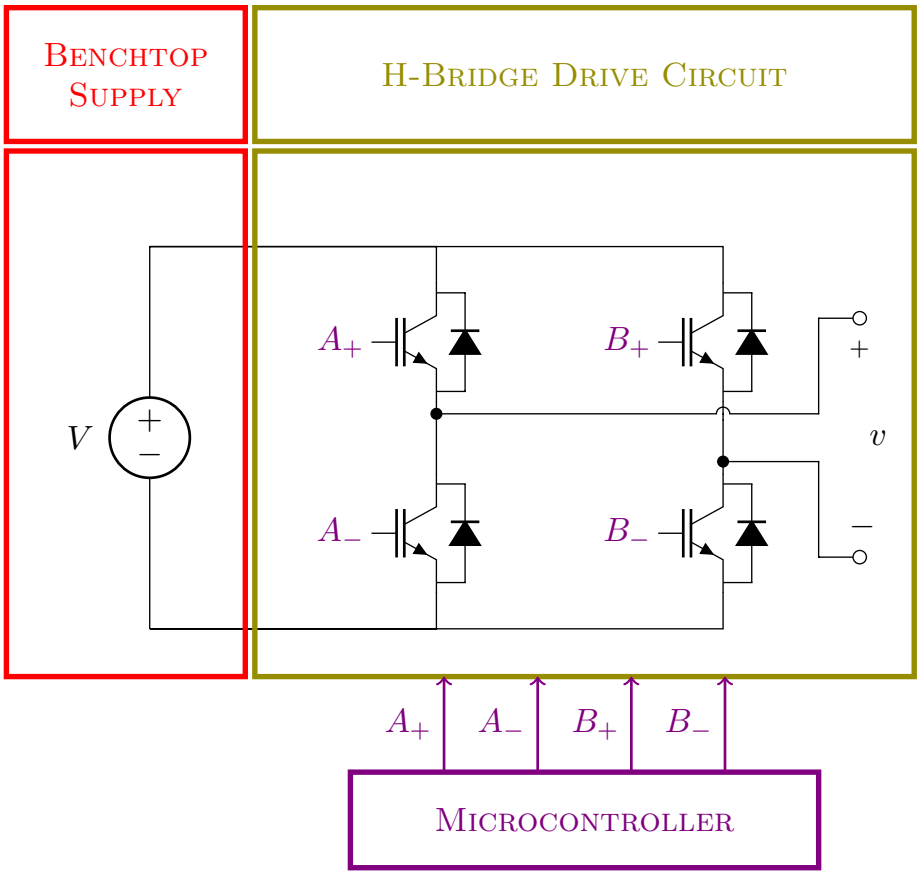


Figure 3.1: An H-bridge circuit controlled by a microcontroller

$2^2 = 4$. The four valid H-bridge configurations are listed in Table 3.1. For each valid configuration, the instantaneous value of logic signals A_+, A_-, B_+, and B_- is provided, along with the resulting instantaneous value of the H-bridge output voltage, v. Clearly, an H-bridge can be commanded to produce an output voltage $v \in \{\pm V, 0\}$ *at any time instant* by manipulation of logic signals A_+, A_-, B_+, and B_- using a microcontroller, as needed for the realization task.

Table 3.1: Valid configurations of H-bridge control signals and resulting output

A_+	A_-	B_+	B_-	v
0	1	0	1	0
0	1	1	0	$-V$
1	0	0	1	$+V$
1	0	1	0	0

3.2 Realization Using a Timer and GPIO Functionality

This section describes a naive realization methodology and its drawbacks in order to motivate the need for a more exotic realization method called multi-mode PWM. Most modern microcontrollers have the following features:

1. *General-purpose input/output (GPIO) pins* - digital signal pins on the microcontroller package that can be electrically connected to other devices. The state of these pins, when operated as outputs, can be manipulated by the user (at will) by executing a short set of instructions.

2. *Hardware timers* - digital counter circuits that increment at a programmable rate. These devices can be configured to generate *interrupt events* periodically in time with a user-selectable period.

3. *Interrupt handling* - a mechanism which allows for a user-provided set of instructions, called an interrupt service routine (ISR), to be executed in response to an interrupt event.

With knowledge of these features, and time-optimal control signal $u(\tau)$, one might be tempted to take an intuitive approach to microcontroller-based realization of the

associated signal v, as shown in Algorithm 3. Since the signal v must be produced in real-time, the relationship:

$$Vu\left(\frac{\beta t}{\pi}\right) = v\left(\frac{\beta t}{\pi}\right) \tag{3.2}$$

will be leveraged. This relationship follows from Equation 2.3 and Equation 2.4.

Algorithm 3 Realization of $v\left(\frac{\beta t}{\pi}\right)$ using Timer and GPIO Functionality

```
Configure four GPIO pins in output mode
Assign one GPIO pin to each of A+, A−, B+, B−
Configure a timer to generate an interrupt every Δt seconds
k ← 0
function TIMER_ISR( )                ▷ This ISR executes once every Δt seconds
    v_k ← Vu(βkΔt/π)
    Determine A+, A−, B+, B− using v_k and Table 3.1
    Change the state of the four GPIO pins based on A+, A−, B+, B−
    k ← k + 1;
    return
end function
```

The realization approach of Algorithm 3 is simple, but may be impractical. In order to faithfully realize the signal v, one must choose the *sampling period* Δ_t to be very small. Since the routine TIMER_ISR() is called once every Δ_t seconds, the instructions inside must execute in *under* Δ_t seconds. The table lookup, GPIO manipulation, and counter increment operations require only a few CPU clock cycles to execute. However, computing the *sample value* v_k on each visit to TIMER_ISR() can be computationally expensive, effectively lower-bounding the user's choice of Δ_t. Some options for computing v_k are listed below:

- Compute the sample value v_k on each visit to the ISR using Equation 3.2 and Equation 2.8. This approach is computationally expensive, since the sine function must be evaluated, but has very low memory demands.

- Pre-compute a lookup table containing many samples of the signal v using a desired sampling period Δ_t. This approach is computationally inexpensive,

requiring only a table lookup to produce sample value v_k on each visit to the ISR, but has very significant memory demands.

- Compute sample value v_k on each visit to the ISR using Equation 3.2 and Equation 2.8, but implement the sine function using a combination of a lookup table and interpolation. This approach balances computation and memory usage, but also introduces an approximation to the sine function.

In short, the amount of computation required on each visit to TIMER_ISR() depends on the amount of available memory on board the microcontroller, and the user's willingness to accept some approximation. If using a low-cost microcontroller with a low CPU clock frequency, no hardware support for floating-point arithmetic operations, and little onboard memory, the lower bound on Δ_t can be quite significant, ultimately limiting the 'quality' of the resulting realization.

The timer/GPIO approach to realization leverages the bang-bang nature of control signals u and v. However, the temporal structure in these signals is not exploited fully. In multi-mode PWM, the *special temporal structure* in signals u and v is exploited to enable the use of PWM hardware for realization. The multi-mode PWM realization approach is admittedly more complex than the timer/GPIO approach, but the added complexity leads to a practically favorable solution which does not require expensive online computation, large memory allocations, or the periodic execution of interrupt service routines at high rates.

3.3 Operation of a PWM Module

Most modern microcontrollers contain one or more pulse-width modulation (PWM) modules. For the purpose of this thesis, a *PWM module* is viewed as a programmable piece of digital hardware, which is comprised of a *counter*, two *compare registers*, and four *logic signal outputs*. A microcontroller typically contains multiple PWM

modules, each with a counter and two compare registers, but each module typically has only two logic signal outputs. However, the various PWM modules in a microcontroller can be synchronized and programmed to operate in unison. Therefore, we imagine the existence of a single PWM module with four logic signal outputs. In this section, the capabilities of a typical PWM module are briefly reviewed.

The basic operating principle of a PWM module is illustrated in Figure 3.2. The module is assumed to have four logic signal outputs, A_+, A_-, B_+, and B_-, which are assumed to drive an H-bridge as shown in Figure 3.1.

In the so-called 'up-count' mode, once the PWM module is powered on, the counter begins to increment at a programmed rate, from zero up to a programmed maximum value. The counter resets to zero when the maximum value is reached, and counts up again. The value of the counter is therefore periodic in time - denote the *PWM counter period* by T_{pwm}. The sawtooth wave c appearing in the top panel of Figure 3.2 represents the value of the counter over time, normalized by the maximum value of the counter. This representation is idealized slightly, but sufficient for illustration purposes; a true plot of the counter's value over time would resemble a staircase.

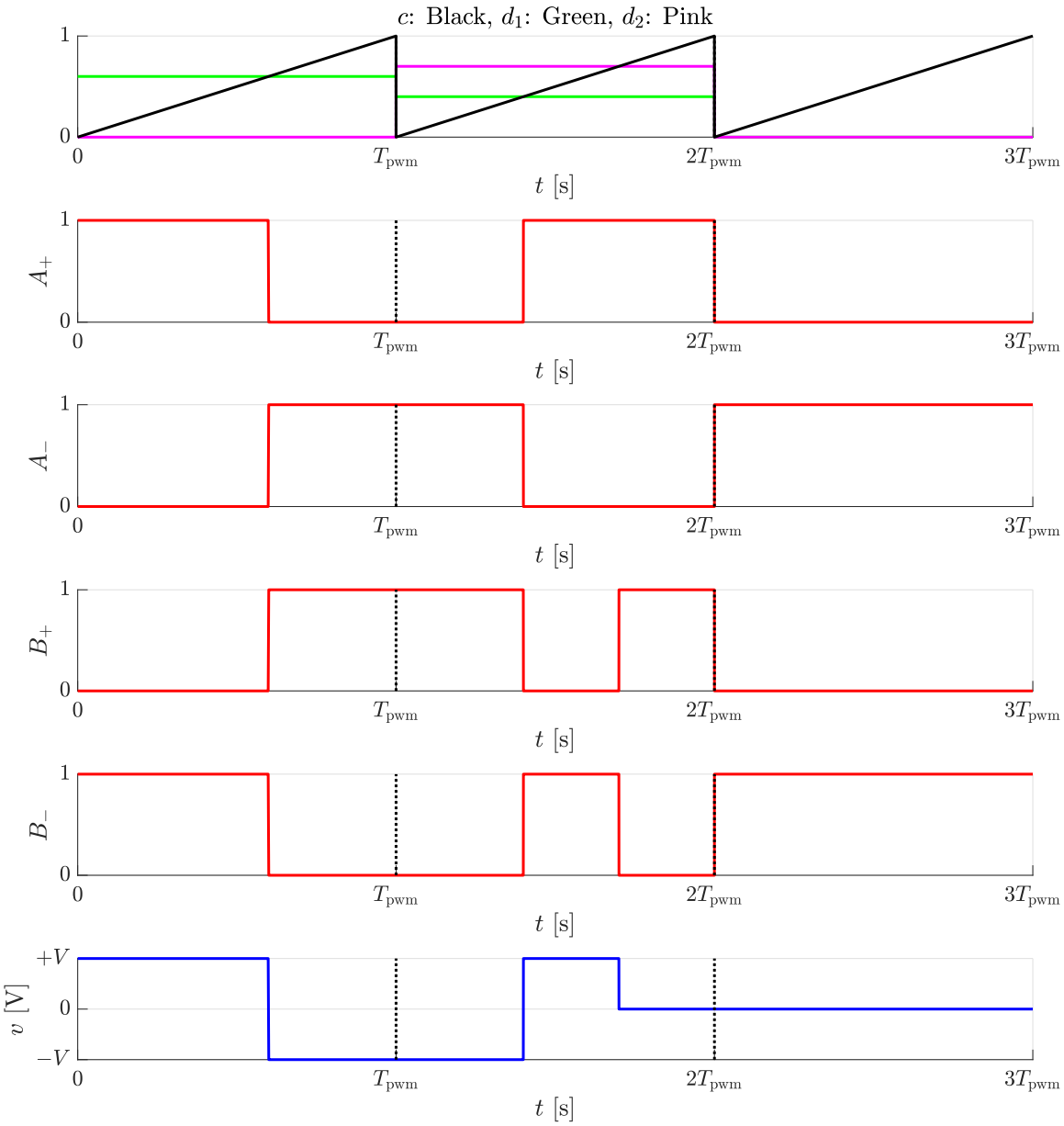


Figure 3.2: Synthesis of H-bridge control and output signals

The two programmable compare registers within the PWM module store a number between zero and the maximum value of the counter, plus one (inclusive). In the top panel of Figure 3.2, d_1 and d_2 represent the values stored in these two registers, normalized by the maximum value of the counter. The values stored in the compare registers can be updated once every T_{pwm} seconds, as shown in top panel of Figure 3.2.

The behavior of each logic signal output of a PWM module is also programmable

and can be configured independently. The logic signal outputs can be configured to change state in response to certain counter-related events. A logic signal output can be programmed to respond to one or more counter-related events, or to ignore one or more counter-related events. Examples of counter-related events are $c = 0$, $c = d_1$, $c = d_2$, and $c = 1$. The response of logic signal outputs to counter-related events can be specified once every T_{pwm} seconds.

For demonstration purposes, the behavior of logic signal outputs A_+ and B_+ is defined as follows in Figure 3.2:

- On the interval $0 \leq t < T_{\text{pwm}}$:

  A_+ is forced high when $c = 0$ and forced low when $c = d_1$

  B_+ is forced low when $c = 0$ and forced high when $c = d_1$

- On the interval $T_{\text{pwm}} \leq t < 2T_{\text{pwm}}$:

  A_+ is forced low when $c = 0$ and forced high when $c = d_1$

  B_+ is forced high when $c = 0$ or $c = d_2$ and forced low when $c = d_1$

- On the interval $2T_{\text{pwm}} \leq t < 3T_{\text{pwm}}$:

  A_+ is forced low when $c = 0$

  B_+ is forced low when $c = 0$

Since A_+, A_-, B_+, and B_- are assumed to drive an H-bridge, the behavior of signals A_- and B_- is constrained by Equation 3.1. The H-bridge output signal v is plotted in the bottom panel of Figure 3.2. Output waveform v may be viewed as a *sequence* of 'building-block waveforms', each T_{pwm} seconds in duration. Note that the features of the PWM module allow for the behavior of each 'building-block waveform' to be independently configured. Each 'building-block waveform' can exhibit zero, one, or two *edges* (changes in value), but no more, since there are only two compare

registers. The placement of edges is controlled by setting compare register values, or equivalently, d_1 and d_2.

3.4 Multi-Mode PWM Decomposition

3.4.1 Groundwork

In Chapter 2, CP1 was solved to yield the signal $u(\tau)$ which is parameterized by four numbers, ψ_1, $\tau_{f,1}$, ψ_2, and $\tau_{f,2}$, as shown in Equation 2.8. In this section, the special temporal structure of $u(\tau)$ will be exploited to decompose it into a sequence of primitive, wavelet-like building blocks called *mode waveforms*. The representation of $u(\tau)$ as a sequence of mode waveforms is referred to as the multi-mode PWM decomposition of $u(\tau)$. Given $u(\tau)$, computing its multi-mode PWM decomposition involves performing only simple arithmetic operations using ψ_1, $\tau_{f,1}$, ψ_2, and $\tau_{f,2}$.

Ten distinct mode waveforms will be defined. All ten mode waveforms share three defining characteristics, listed below. These characteristics of the mode waveforms make it possible for them to be generated using PWM hardware on a typical microcontroller and an H-bridge, as detailed later in this chapter:

1. All ten mode waveforms are T_{pwm} seconds in duration
2. All ten mode waveforms take values in $\{\pm V, 0\}$ at any point in time
3. No mode waveform contains more than two *edges* (changes in value)

A PWM module with the capabilities described in the previous section is capable of producing 21 distinct waveforms that meet the above criteria. However, only 10 of these waveforms are needed in order to realize time optimal control signals for second order oscillators.

Computing the multi-mode PWM decomposition of $u(\tau)$ begins with the compu-

tation of six quantities, $\tau_{1,1}$, N_1, $\tau_{2,1}$, $\tau_{1,2}$, N_2, and $\tau_{2,2}$, according to:

$$\tau_{1,i} := \begin{cases} 1 - \psi_i/\pi & , 0 < \psi_i \leq \pi \\ 2 - \psi_i/\pi & , \pi < \psi_i \leq 2\pi \end{cases} \qquad , i \in \{1, 2\} \tag{3.3a}$$

$$N_i := \text{floor}\left(\tau_{f,i} - \tau_{1,i}\right) \qquad , i \in \{1, 2\} \tag{3.3b}$$

$$\tau_{2,i} := \tau_{f,i} - N_i - \tau_{1,i} \qquad , i \in \{1, 2\} \tag{3.3c}$$

Note that $\tau_{1,i} \in [0, 1)$, $\tau_{2,i} \in [0, 1)$, and N_i is a nonnegative integer.

The six quantities calculated using Equation 3.3 may be interpreted as attributes of $u(\tau)$ that describe its shape. Inspection of Equation 2.8 reveals that $u(\tau)$ is piecewise constant, with constant segments no longer than 1 unit of scaled time τ in duration. In fact, $u(\tau)$ contains $N_1 + N_2$ constant segments of duration 1, and up to four (shorter) constant segments of durations $\tau_{1,1}$, $\tau_{2,1}$, $\tau_{1,2}$, and $\tau_{2,2}$.

The waveform $u(\tau)$ cannot be immediately decomposed into a sequence of mode waveforms. Thus, we apply a specially chosen delay, τ_d, to $u(\tau)$, and decompose the delayed signal $u(\tau - \tau_d)$ into a sequence of mode waveforms. The delay τ_d is computed as:

$$\tau_d = \begin{cases} \text{ceil}(\tau_{f,1}) - \tau_{f,1} & , \tau_{1,1} + \tau_{2,1} \geq 1 \\ \text{ceil}(\tau_{f,1}) - \tau_{f,1} + 1 & , \tau_{1,1} + \tau_{2,1} < 1 \end{cases}$$

The insertion of delay does not influence the shape of the oscillator's response to the control (due to a modeling assumption of time invariance), or the time optimality of the solution. *By design*, the delayed signal $u(\tau - \tau_d)$ has the following properties:

- has *no more than two* edges on any interval $\tau \in (k, k+1), k \in \mathbb{N}$
- always reaches a final value of zero before $\tau = N_1 + N_2 + 4$
- can be expressed as a sequence of $N_1 + N_2 + 4$ mode waveforms, each 1 unit of

scaled time τ in duration

Recall that mode waveforms were assumed to have no more than two edges within a span of T_{pwm} seconds, or equivalently, within $(\beta/\pi)T_{\text{pwm}}$ units of scaled time τ. Close inspection of Equation 2.8 reveals that $u(\tau)$ contains no more than two edges within a span of 1 unit of scaled time τ, except in one special case. The special case arises if $\tau_{2,1} + \tau_{1,2} < 1$, and if $u(\tau)$ contains an edge at $\tau = \tau_{f,1}$, resulting in the presence of three edges within a span of 1 unit of scaled time τ. In this case, the delayed signal $u(\tau - \tau_d)$ has an edge at $\tau = \tau_{f,1} + \tau_d$ which is *aligned to an integer value of* τ *by definition of* τ_d. *It follows that* $u(\tau - \tau_d)$ *has no more than two edges between any two successive integer values of* τ. *Consequently, by defining the mode waveforms to have a duration of 1 unit in scaled time* τ, *or equivalently, by defining:*

$$T_{\text{pwm}} := \frac{\pi}{\beta} \text{ seconds} \tag{3.4}$$

then $u(\tau - \tau_d)$ *can be expressed as a sequence of mode waveforms.*

It is helpful to visualize the notation and concepts introduced thus far using a concrete example. A plot of $u(\tau - \tau_d)$ is provided in Figure 3.3 for the particular case (chosen arbitrarily) where $\psi_1 = \frac{5\pi}{3}$, $\tau_{f,1} = 5.2$, $\psi_2 = \frac{5\pi}{3}$, and $\tau_{f,2} = 3.9$. Note that $u(\tau - \tau_d)$ is nonzero on $\tau \in [\tau_d, \tau_d + \tau_{f,1} + \tau_{f,2})$, but is plotted on the wider interval $\tau \in [0, N_1 + N_2 + 4]$.

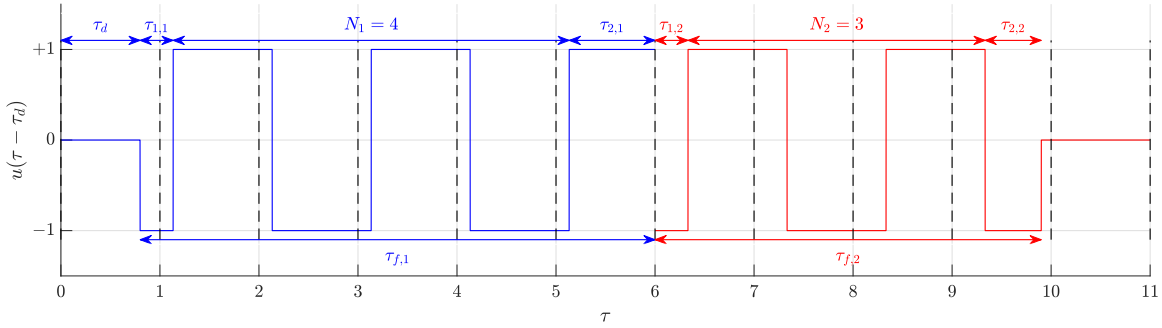


Figure 3.3: Visual depiction of quantities $\tau_{f,i}$, $\tau_{1,i}$, N_i, $\tau_{2,i}$, and τ_d.

Time instants where scaled time τ is an integer are marked with dashed black lines in Figure 3.3. As expected, there are no more than two edges in $u(\tau - \tau_d)$ between any two successive dashed black lines. An additional property of $u(\tau - \tau_d)$ can be observed in Figure 3.3. Although not a requirement for computing a multi-mode PWM decomposition, it is generally true that only the intervals $\tau \in (1, 2)$ and $\tau \in (N_1 + N_2 + 2, N_1 + N_2 + 3)$ can contain two edges.

3.4.2 Decomposition Procedure

Ten mode waveforms are needed to account for the full range of possible behaviors in $u(\tau)$ on $\tau \in (k, k+1), k \in \mathbb{N}$. *The ten requisite mode waveforms are defined graphically in Figure 3.4.* Notationally, we use m to index modes, and $\nu_m(t \mid d_1, d_2)$ to refer to the mode waveform associated to mode m. The mode waveforms are presented as functions of time t (with units of seconds), as opposed to scaled time τ to emphasize that they are generated at the output of an H-bridge in the real world.

Each mode waveform has a duration of $T_{\text{pwm}} = \pi/\beta$ seconds (equivalently, 1 unit of scaled time τ). The mode waveforms associated with modes $m = 5, 6, 9, 10$ are characterized by two timing parameters, $d_1, d_2 \in [0, 1]$, with $d_2 > d_1$, whereas the remaining mode waveforms are characterized only by a single timing parameter, $d_1 \in [0, 1]$. Note that for $m \neq 5, 6, 9, 10$, timing parameter d_2 appears in the notation describing the associated mode waveform, but does not actually influence the behavior of a mode waveform. For example, the mode waveform for mode $m = 1$ is denoted by $\nu_1(t \mid d_1, d_2)$, but timing parameter d_2 is not needed to generate $\nu_1(t \mid d_1, d_2)$. This is done for notational convenience later on. An eleventh mode, $m = 0$, is not shown. Its associated mode waveform, $\nu_0(t \mid d_1, d_2)$, is zero for $t \in (0, T_{\text{pwm}})$. The behavior of $\nu_0(t \mid d_1, d_2)$ is not influenced by timing parameters d_1 or d_2. We will also assume for convenience that the waveform $\nu_m(t \mid d_1, d_2)$ is zero for all $t \notin (0, T_{\text{pwm}})$, for $m = 0, 1, \ldots 10$.

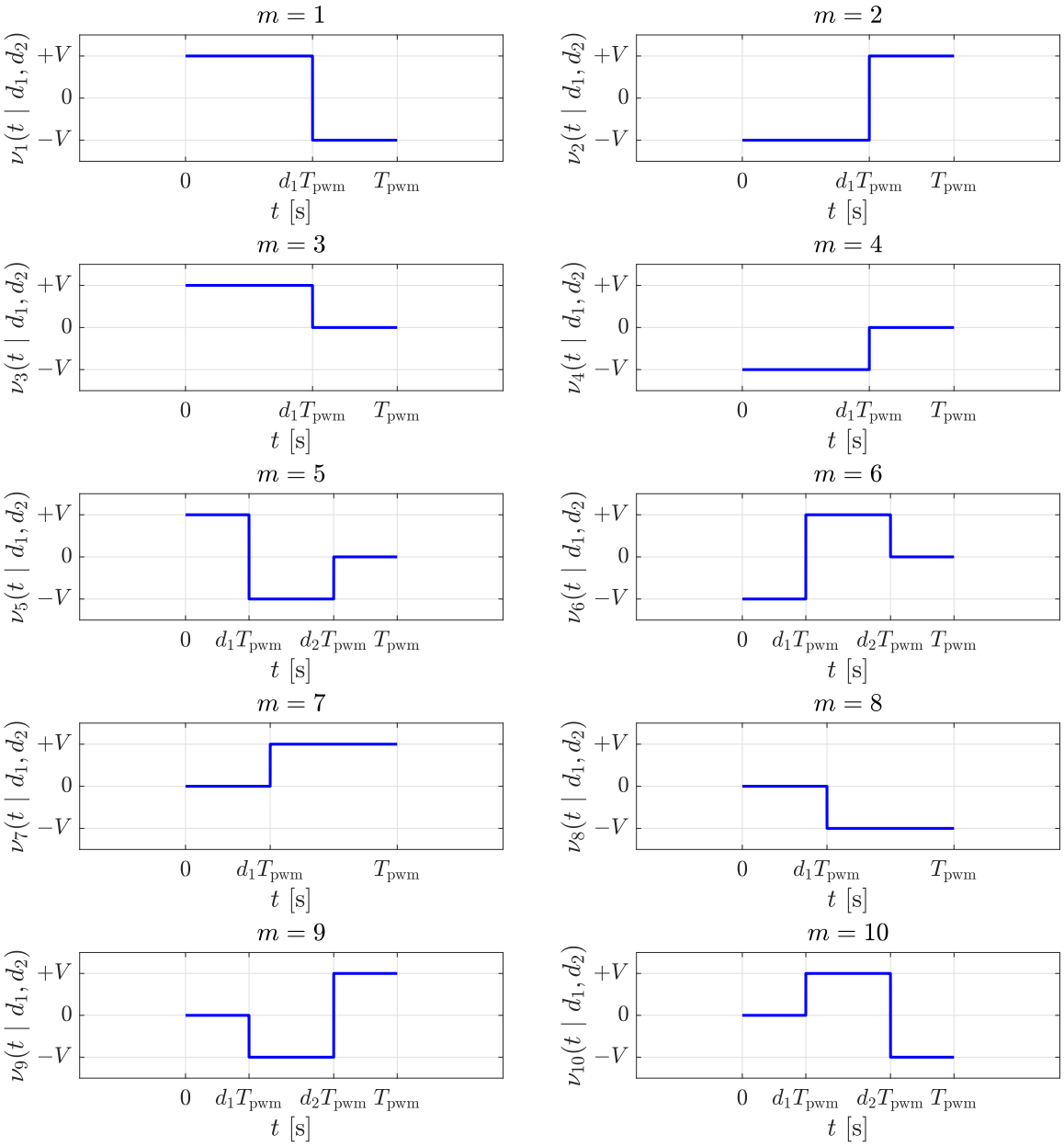


Figure 3.4: Graphical definition of mode waveforms $\nu_m(t \mid d_1, d_2)$ for $m = 1, \ldots, 10$

The analysis in the previous subsection reveals that $u(\tau - \tau_d)$, and thus, $v(\tau - \tau_d)$, can be expressed as a sequence of $N_1 + N_2 + 4$ mode waveforms, each 1 unit of scaled time τ in duration. Each mode waveform in the sequence is characterized by a scalar mode index, m, and up to two scalar timing parameters, d_1 and d_2, as shown in Figure 3.4. It follows that the multi-mode PWM decomposition of $u(\tau - \tau_d)$ (or $v(\tau - \tau_d)$) is characterized by three lists of $N_1 + N_2 + 4$ scalars. Thus, we define three

arrays of length N_1+N_2+4 to store the multi-mode PWM decomposition of $u(\tau-\tau_d)$ (or $v(\tau-\tau_d)$):

- Array $\boldsymbol{m}$: the k^{th} entry of this array, denoted $\boldsymbol{m}[k]$, contains a scalar mode index $m \in \{0,1,\ldots,10\}$ which identifies the mode waveform which (after time scaling) describes the behavior of $u(\tau-\tau_d)$ (or $v(\tau-\tau_d)$) on the interval $k \leq \tau < k+1$.

- Array $\boldsymbol{d}_1$: the k^{th} entry of this array, denoted $\boldsymbol{d}_1[k]$, contains the scalar timing parameter d_1 associated with the mode waveform identified by $\boldsymbol{m}[k]$.

- Array $\boldsymbol{d}_2$: the k^{th} entry of this array, denoted $\boldsymbol{d}_2[k]$, contains the scalar timing parameter d_2 associated with the mode waveform identified by $\boldsymbol{m}[k]$.

It follows that for $k=0,1,\ldots,N_1+N_2+3$, one can express $v(\tau-\tau_d)=Vu(\tau-\tau_d)$ on the interval $k \leq \tau < k+1$ using the mode waveforms of Figure 3.4 and the triple $(\boldsymbol{m}[k],\boldsymbol{d}_1[k],\boldsymbol{d}_2[k])$ according to:

$$v(\tau-\tau_d)=Vu(\tau-\tau_d)=\nu_{\boldsymbol{m}[k]}\left(\frac{\pi}{\beta}(\tau-k)\;\middle|\;\boldsymbol{d}_1[k],\boldsymbol{d}_2[k]\right) \quad ,k\leq\tau<k+1 \tag{3.5}$$

We now provide explicit formulas for computing the triple $(\boldsymbol{m}[k],\boldsymbol{d}_1[k],\boldsymbol{d}_2[k])$ which encodes the behavior of $v(\tau-\tau_d)$ on the interval $k \leq \tau < k+1$. Only the four quantities ψ_1, $\tau_{f,1}$, ψ_2, and $\tau_{f,2}$ (known from the solution to CP1) are needed to determine $(\boldsymbol{m}[k],\boldsymbol{d}_1[k],\boldsymbol{d}_2[k])$:

$$\boldsymbol{m}[k] := \begin{cases} \boldsymbol{m}_1[k] & ,k=0,\ldots,N_1+1 \\ \boldsymbol{m}_2[k-(N_1+2)] & ,k=N_1+2,\ldots,N_1+N_2+3 \end{cases} \tag{3.6}$$

$$\boldsymbol{d}_1[k] := \begin{cases} \boldsymbol{d}_{1,1}[k] & ,k=0,\ldots,N_1+1 \\ \boldsymbol{d}_{1,2}[k-(N_1+2)] & ,k=N_1+2,\ldots,N_1+N_2+3 \end{cases} \tag{3.7}$$

$$\boldsymbol{d}_2[k] := \begin{cases} \boldsymbol{d}_{2,1}[k] & ,k=0,\ldots,N_1+1 \\ \boldsymbol{d}_{2,2}[k-(N_1+2)] & ,k=N_1+2,\ldots,N_1+N_2+3 \end{cases} \tag{3.8}$$

where $\boldsymbol{m}_1[k]$ is defined in Table 3.2, $\boldsymbol{m}_2$ is defined in Table 3.3, and:

$$\boldsymbol{d}_{1,1}[k] = \begin{cases} 2 - \tau_{1,1} + \tau_{2,1} & , \boldsymbol{m}_1[k] = 7, 8 \\ 1 - \tau_{1,1} + \tau_{2,1} & , \boldsymbol{m}_1[k] = 9, 10 \\ 1 - \tau_{2,1} & , \text{otherwise} \end{cases} \tag{3.9a}$$

$$\boldsymbol{d}_{2,1}[k] = \begin{cases} 1 - \tau_{2,1} & , \boldsymbol{m}_1[k] = 9, 10 \\ \text{unused, set arbitrarily} & , \text{otherwise} \end{cases} \tag{3.9b}$$

$$\boldsymbol{d}_{1,2}[k] = \begin{cases} \tau_{1,2} + \tau_{2,2} - 1 & , \boldsymbol{m}_2[k] = 3, 4 \\ \tau_{1,2} & , \text{otherwise} \end{cases} \tag{3.10a}$$

$$\boldsymbol{d}_{2,2}[k] = \begin{cases} \tau_{1,2} + \tau_{2,2} & , \boldsymbol{m}_2[k] = 5, 6 \\ \text{unused, set arbitrarily} & , \text{otherwise} \end{cases} \tag{3.10b}$$

Table 3.2: Elementwise determination of mode array $\boldsymbol{m}_1$

$\psi_1 \leq \pi$?	N_1 Odd?	$\tau_{1,1} + \tau_{2,1} \leq 1$?	$\boldsymbol{m}_1[0]$	$\boldsymbol{m}_1[1]$	$\boldsymbol{m}_1[k]$ $k = 2 \ldots N_1 + 1$
0	0	0	8	2	$1, 2, 1, 2, \ldots$
0	0	1	0	9	$1, 2, 1, 2, \ldots$
0	1	0	8	2	$1, 2, 1, 2, \ldots$
0	1	1	0	9	$1, 2, 1, 2, \ldots$
1	0	0	7	1	$2, 1, 2, 1, \ldots$
1	0	1	0	10	$2, 1, 2, 1, \ldots$
1	1	0	7	1	$2, 1, 2, 1, \ldots$
1	1	1	0	10	$2, 1, 2, 1, \ldots$

Table 3.3: Elementwise determination of mode array $\boldsymbol{m}_2$

$\psi_2 \leq \pi$?	N_2 Odd?	$\tau_{1,2} + \tau_{2,2} \leq 1$?	$\boldsymbol{m}_2[k]$ $k = 0 \ldots N_2 - 1$	$\boldsymbol{m}_2[N_2]$	$\boldsymbol{m}_2[N_2 + 1]$
0	0	0	$2, 1, 2, 1, \ldots$	2	3
0	0	1	$2, 1, 2, 1, \ldots$	6	0
0	1	0	$2, 1, 2, 1, \ldots$	1	4
0	1	1	$2, 1, 2, 1, \ldots$	5	0
1	0	0	$1, 2, 1, 2, \ldots$	1	4
1	0	1	$1, 2, 1, 2, \ldots$	5	0
1	1	0	$1, 2, 1, 2, \ldots$	2	3
1	1	1	$1, 2, 1, 2, \ldots$	6	0

Once arrays $\boldsymbol{m}$, $\boldsymbol{d}_1$, and $\boldsymbol{d}_2$ are populated according to Equation 3.6, Equation 3.9, and Equation 3.10, the multi-mode PWM decomposition of $u(\tau - \tau_d)$ (or $v(\tau - \tau_d)$) is fully specified. That is, for $k = 0, 1, \ldots, N_1 + N_2 + 3$, one can express the delayed plant input signal $v(\tau - \tau_d)$ on the interval $k \leq \tau < k+1$ using the mode waveforms of Figure 3.4 and the triple $(\boldsymbol{m}[k], \boldsymbol{d}_1[k], \boldsymbol{d}_2[k])$ according to Equation 3.5. Consequently, one can invoke Equation 2.3 and Equation 2.4 to write:

$$Vu(\tau - \tau_d) = v\left(\frac{\beta(t - t_d)}{\pi}\right)$$

where $t_d := \frac{\pi \tau_d}{\beta}$, and subsequently express the delayed signal v as a function of time in seconds, t, in terms of the mode waveforms of Figure 3.4 and the entries of $\boldsymbol{m}$, $\boldsymbol{d}_1$, and $\boldsymbol{d}_2$ as:

$$v\left(\frac{\beta(t - t_d)}{\pi}\right) = \sum_{k=0}^{N_1+N_2+3} \nu_{\boldsymbol{m}[k]}\left(t - \frac{\pi}{\beta}k \,\middle|\, \boldsymbol{d}_1[k], \boldsymbol{d}_2[k]\right) \tag{3.11}$$

Note that there is no overlap in time between successive terms in the summation, since all mode waveforms ($\nu_m(t \mid d_1, d_2)$ for $m = 0, \ldots 10$) are assumed to be zero-valued outside of $t \in [0, \frac{\pi}{\beta}]$.

3.5 Microcontroller Programming Details

In order to realize the delayed signal v in real-time (i.e. as a function of t) using multi-mode PWM and the hardware setup in Figure 3.1, it is necessary to program a microcontroller. First, low-level code must be developed to command the onboard PWM module to generate logic signals A_+, A_-, B_+, and B_- in such a way that a given mode waveform $\nu_m(t \mid d_1, d_2)$ appears at the H-bridge output over a period of $T_{\text{pwm}} = \pi/\beta$ seconds. Then, higher-level code must be developed (which uses the low-level code) to produce a particular sequence of $N_1 + N_2 + 4$ back-to-back mode waveforms at the H-bridge output.

3.5.1 Low-Level PWM Module Programming

The programmable features and operation of a typical PWM module are described in Section 3.3. In this subsection, we state how each mode waveform (T_{pwm} seconds in duration) shown in Figure 3.4 can be produced at the output of an H-bridge which is driven by four logic signals, A_+, A_-, B_+, and B_-, sourced from a PWM module. The relationship between the four logic signals is given by Equation 3.1. Thus, only the responses of two (out of the four) logic signals to counter-related events must be specified to fully determine the behavior of the H-bridge output signal over a period of T_{pwm} seconds. Table 3.4 reveals how to program the responses of logic signals A_+ and B_+ to counter-related events $c = 0$, $c = d_1$, and $c = d_2$ in order to produce a desired mode waveform $\nu_m(t \mid d_1, d_2)$ at the H-bridge output, for $m = 0, \ldots, 10$. When a counter-related event occurs, each output signal can be configured to respond by forcing high (), forcing low (), or to not respond and maintain its value

(———). Of course, the PWM module's compare registers must also be populated based on values of d_1 and d_2.

Implementation of this strategy on a particular microcontroller requires mastering the low-level programming details specific to that device. With this knowledge, one could implement a routine called `configureSwitchBehavior(`m`, `d_1`, `d_2`)`, which sets the compare registers within a PWM module using user-provided values of d_1 and d_2, in addition to configuring the response of logic signal outputs A_+, A_-, B_+, and B_- according to Table 3.4 and Equation 3.1. The routine `configureSwitchBehavior(`m`, `d_1`, `d_2`)` takes control of the H-bridge output for T_{pwm} seconds after it is called, and produces the waveform $\nu_m(t \mid d_1, d_2)$ at the H-bridge output during this time.

Table 3.4: Response of A_+ and B_+ to counter events for mode waveform realization

	Response of A_+			Response of B_+		
Mode	$c = 0$	$c = d_1$	$c = d_2$	$c = 0$	$c = d_1$	$c = d_2$
$m = 0$						
$m = 1$						
$m = 2$						
$m = 3$						
$m = 4$						
$m = 5$						
$m = 6$						
$m = 7$						
$m = 8$						
$m = 9$						
$m = 10$						

3.5.2 High-Level Microcontroller Programming

The delayed time-optimal control signal v is realized using multi-mode PWM and applied to the plant in open-loop fashion. Thus, arrays $\boldsymbol{m}$, $\boldsymbol{d}_1$, and $\boldsymbol{d}_2$ can be populated immediately after CP1 is solved, and stored in lookup tables prior to exciting the plant. It is clear from Equation 3.11 that starting at $t = 0$, the routine `configureSwitchBehavior(`m`, `d_1`, `d_2`)` must be called every $T_{\text{pwm}} = \frac{\pi}{\beta}$ seconds to produce a new mode waveform at the H-bridge output, until all $N_1 + N_2 + 4$ required mode waveforms have been produced. Prior to each call, arrays $\boldsymbol{m}$, $\boldsymbol{d}_1$, and $\boldsymbol{d}_2$ must be accessed to obtain the three scalar input arguments m, d_1, and d_2. An interrupt-based programming approach is the natural choice to ensure that `configureSwitchBehavior(`m`, `d_1`, `d_2`)` is called in a time-periodic manner. Instead of using a hardware timer to generate interrupt events as in Section 3.2, the PWM module can be configured to generate time-periodic interrupt events. For instance, an interrupt event can be generated in response to the counter-related event $c = 0$, which occurs once every T_{pwm} seconds. Using an interrupt service routine, the entries of arrays $\boldsymbol{m}$, $\boldsymbol{d}_1$, and $\boldsymbol{d}_2$ can be accessed every T_{pwm} seconds and used to manipulate the behavior of the PWM module, as shown in Algorithm 4.

Algorithm 4 Realization of $v\left(\frac{\beta(t-t_d)}{\pi}\right)$ using PWM Hardware

```
Solve CP1
Populate arrays m, d1, and d2
Set a PWM counter period of T_pwm = π/β seconds
Configure PWM module to generate an interrupt every T_pwm seconds
k ← 0
function PWM_ISR( )                ▷ This ISR executes once every T_pwm seconds
    m ← m[k]; (Table Lookup)
    d1 ← d1[k]; (Table Lookup)
    d2 ← d2[k]; (Table Lookup)
    configureSwitchBehavior(m, d1, d2);
    k ← k + 1;
    return
end function
```

The benefit of the lookup table based implementation is that very little code needs to be executed inside PWM_ISR() in real time. The arrays $\boldsymbol{m}$, $\boldsymbol{d}_1$, and $\boldsymbol{d}_2$ simply need to be accessed, and the appropriate PWM mode and timing parameters contained in them must be imposed on a cycle-by-cycle basis by writing to registers. Unless the damped resonant frequency of the oscillator under study (β) is very large, we can be assured that PWM_ISR() will terminate within T_{pwm} seconds due to the simple nature of the instructions being executed.

3.5.3 Implementation Considerations

In order to realize the delayed time-optimal control signal v using multi-mode PWM, the PWM hardware module must meet the following requirements:

1. It is possible to set a PWM counter period of $T_{\text{pwm}} = \frac{\pi}{\beta}$ seconds
2. It is possible to generate *two* switches in the PWM waveform per switching cycle, at arbitrary points in time
3. It is possible to change PWM mode and timing parameters on a cycle-by-cycle basis

Requirement 1 may or may not be met, depending on the damped resonant frequency of the oscillator to be controlled (β), datasheet-specified limits on T_{pwm}, and quantization-related effects, briefly described below. With an exact solution to CP1 and an ideal H-bridge circuit, the proposed multi-mode PWM scheme described in this chapter *exactly* produces the waveform given in Equation 3.11. In reality, both the PWM counter period (T_{pwm}) and timing parameters (d_1, d_2) are finely quantized on most microcontrollers, and hence it is not possible to place edges in v at truly arbitrary points in time. The quantization of edge placement is not fixed, and can be manipulated by setting design parameters within the PWM module. However, it is true that this quantization error becomes more severe as the PWM counter period

(T_{pwm}) is decreased. The experimental results in the next chapter indicate that the effect of this quantization is insignificant, at least for the specific microcontroller and second-order oscillator used for experimentation.

Requirement 2 falls within the capabilities of a typical PWM module due to the presence of two compare registers as described in Section 3.3. Requirement 3 is also within the capabilities of a typical microcontroller, but deserves some discussion. In order to achieve a particular behavior out of the PWM module on the time interval $kT_{\text{pwm}} \leq t < (k+1)T_{\text{pwm}}$, one must prepare the appropriate module configuration parameters (i.e. logic signal responses to counter events, d_1, and d_2) and load them into their respective device registers *prior* to time $t = kT_{\text{pwm}}$. However, one also needs to ensure that the module does not respond immediately to the new set of configuration parameters, but rather waits until $t = kT_{\text{pwm}}$ to do so. This module behavior is possible to achieve by making use of a feature called *register shadowing* that is present in most modern microcontrollers.

CHAPTER 4
EXPERIMENTAL VALIDATION

A series RLC circuit was constructed to serve as the stable, damped harmonic oscillator to be controlled. A circuit schematic is shown inside the blue box of Figure 4.1. In principle, any second-order system could be used for validation, so long as it has a conjugate-pair of eigenvalues in the left half of the complex plane. However, the series RLC circuit is advantageous for experimental purposes, as its state variables are easily measured using an oscilloscope, in contrast to other second-order systems, such as the transducer systems. Furthermore, system eigenvalues are completely determined by the choice of circuit component values, which means that a series RLC circuit can serve as a *proxy* for other oscillatory physical systems of order two, for the purposes of control design.

In order to apply our control design methodology, note that the dynamics of this plant are given by Equation 2.2a, where:

$$\alpha = \frac{R}{2L} \quad , \quad \beta = \sqrt{\frac{1}{LC} - \left(\frac{R}{2L}\right)^2} \quad , \quad \gamma = \frac{1}{L} \tag{4.1}$$

and the state variables w_1 and w_2 are as indicated in Figure 4.1. This state-space model can then be transformed into the dimensionless single parameter form of Equation 2.1a by following the procedure described in Section 2.1.

4.1 Experimental Setup

The proposed control strategy was experimentally validated using the hardware shown in Figure 4.1. Discrete components with nominal values $R = 24\ \Omega$, $L = 3.3$ mH, $C = 47$ nF were used to build the series RLC circuit. Based on nominal

component values, we expect our system to be characterized by $\alpha = 3636.36$ rad/s and $\beta = 80213.65$ rad/s (therefore $a = 0.1424$). A Texas Instruments DRV8305 BoosterPack was operated as an H-Bridge to provide the requisite bang-bang input signal to the plant. A bench-top power supply provided $V = 6$ V to the H-bridge.

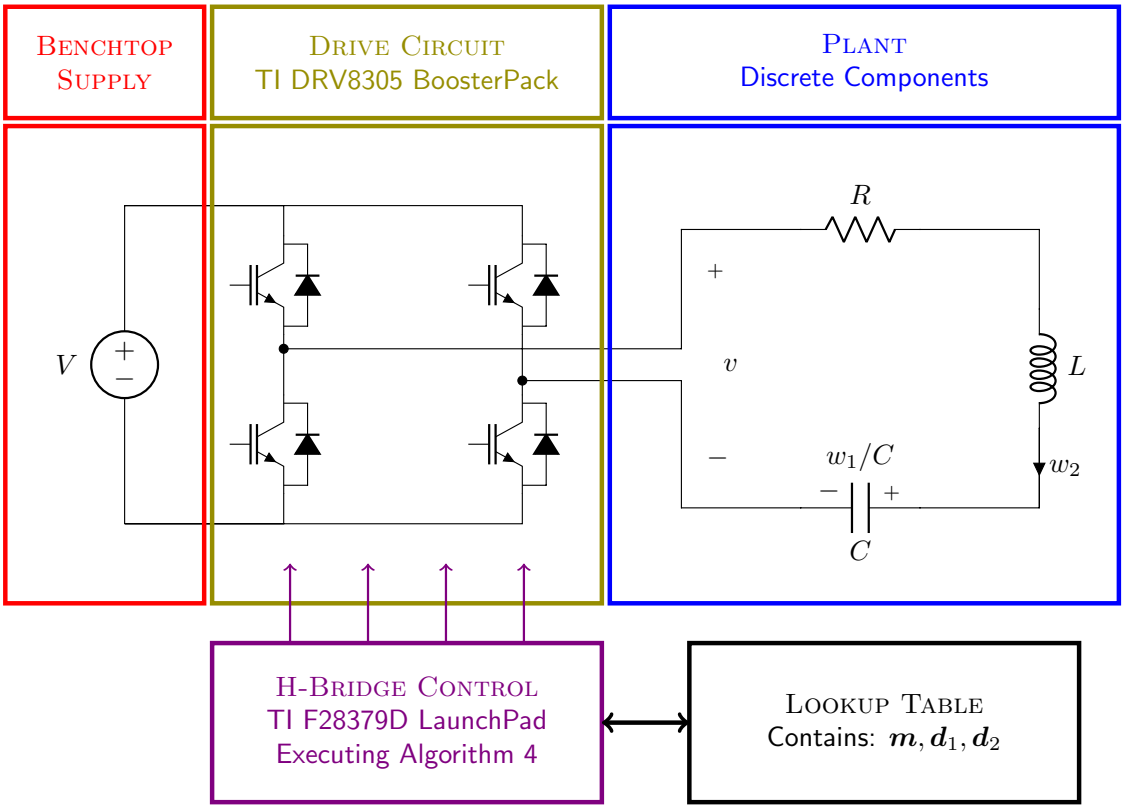


Figure 4.1: Experimental setup used to validate the proposed control design

4.2 Least-Squares Parameter Identification

The proposed control method is open-loop, which means that it is inherently sensitive to model parameter errors. Sensitivity of open-loop time-optimal control to frequency modeling errors is treated thoroughly in [13]. Additionally, robust time-optimal controllers are presented in [13], which are shown to also yield bang-bang control laws. It is worth noting that applying the methods in [13] to improve robustness to frequency modeling error does not impact the ability to realize the time-optimal

control via the multi-mode PWM scheme, as the resulting control is still in the form of Equation 2.7.

The success of an open-loop control scheme is dependent on accurate knowledge of the plant model and its parameters. Thus, time-optimal control design for the series RLC plant was performed using parameters estimated from measured input-output data, as opposed to nominal component values. $M = 742500$ samples of excitation voltage v and loop current y (taken at time instants $t_1, \ldots, t_M$) were used to estimate model parameters. The relationship between loop current y and excitation voltage v in the Laplace domain is given by:

$$\left(s^2 + \frac{R}{L}s + \frac{1}{LC}\right)\mathcal{Y}(s) = \left(\frac{s}{L}\right)\mathcal{V}(s) \tag{4.2}$$

To avoid using numerically approximated derivatives of the measurement data in parameter identification, both sides of Equation 4.2 are processed by a third-order lowpass filter $\mathcal{H}(s) = \left(\frac{\lambda}{s+\lambda}\right)^3$ with bandwidth $\lambda > 0$. After routine manipulations, the following over-constrained system of equations in three unknown parameters may be constructed from processed measurement data:

$$\begin{bmatrix} -y_1(t_1) & -y_0(t_1) & v_1(t_1) \\ \vdots & \vdots & \vdots \\ -y_1(t_M) & -y_0(t_M) & v_1(t_M) \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} y_2(t_1) \\ \vdots \\ y_2(t_M) \end{bmatrix} \tag{4.3}$$

where

$$y_i(t) := \mathcal{L}^{-1}\left\{s^i \mathcal{H}(s)\mathcal{Y}(s)\right\} \qquad , i \in \{0, 1, 2\}$$

$$v_1(t) := \mathcal{L}^{-1}\left\{s\mathcal{H}(s)\mathcal{V}(s)\right\}$$

$$c_1 := \frac{\hat{R}}{\hat{L}} \ , \ c_2 := \frac{1}{\hat{L}\hat{C}} \ , \ c_3 := \frac{1}{\hat{L}}$$

The linear system of equations in Equation 4.3 is then solved in the least-squares sense to obtain c_1, c_2, and c_3, which are then processed to obtain estimates $\hat{R}$, $\hat{L}$, and $\hat{C}$ for the circuit parameters, according to:

$$\hat{R} = \frac{c_1}{c_3} \quad , \quad \hat{L} = \frac{1}{c_3} \quad , \quad \hat{C} = \frac{c_3}{c_2}$$

The mapping between c_1, c_2, and c_3 and estimated circuit parameters $\hat{R}$, $\hat{L}$, and $\hat{C}$ is nonlinear, but one-to-one, so there is no possibility of obtaining multiple solutions or no solution when recovering circuit parameter estimates from c_1, c_2, c_3.

Using the input-output dataset shown in Figure 4.2, the following estimates were obtained for the circuit parameter values:

$$\hat{R} = 35.12\ \Omega \quad , \quad \hat{L} = 3.55\ \text{mH} \quad , \quad \hat{C} = 46.55\ \text{nF}$$

Using Equation 4.1, the resulting estimates for α, β and γ are:

$$\hat{\alpha} = 4948.92\ \text{rad/s} \quad , \quad \hat{\beta} = 77641.83\ \text{rad/s} \quad , \quad \hat{\gamma} = 281.78\ \text{H}^{-1}$$

Thus, our benchtop system is characterized by the parameter $a = 0.2001$. The estimate $\hat{R}$ is much larger than the nominal value the resistor. This is partly due to the series resistance present in the inductor, which was measured to be 7 Ω. To assess the quality of the estimated parameters, a simulation model was constructed using the estimated component values. The response $y_{\text{sim}}(t)$ to *measured* input data was simulated, and has been plotted over measured output data ($y_{\text{mea}}(t)$) in the middle panel of Figure 4.2. The two waveforms are visually indistinguishable, which serves as confirmation of an acceptable model and input realization method. The error signal $e(t) = y_{\text{mea}}(t) - y_{\text{sim}}(t)$ is computed to quantify the goodness-of-fit. $e(t)$ is two orders of magnitude lower in amplitude than $y_{\text{mea}}(t)$. The energy in $e(t)$ is computed to be

0.082% of the energy in $y_{\text{mea}}(t)$.

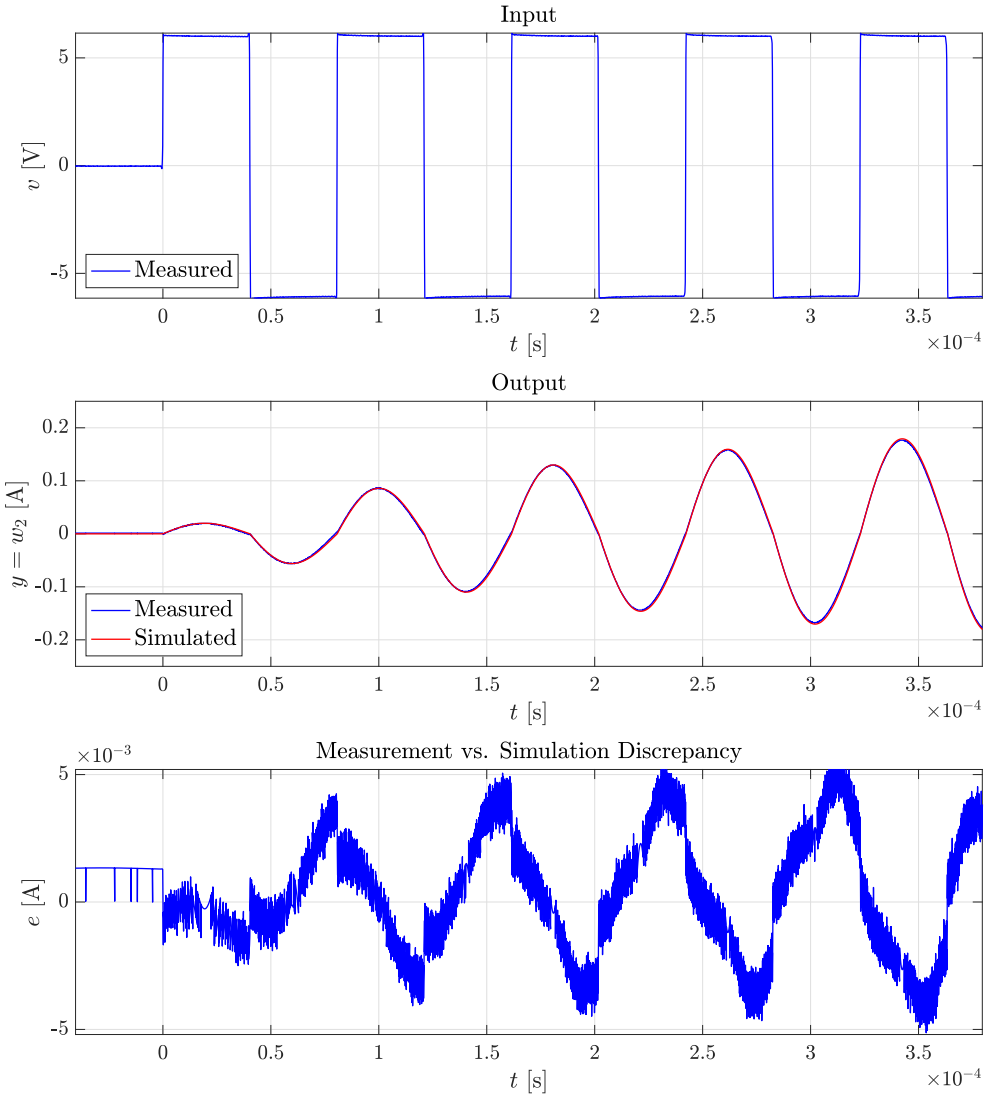


Figure 4.2: Parameter identification dataset and fit results

4.3 Experimental Results

After learning the model parameters of our second-order oscillator via least-squares parameter identification, we proceed to formulate and solve CP1. We define the plant output in physical state variables as in Equation 2.2b, that is: $y = w_2$. Consequently, the plant output equation used in the solution of CP1 is given by Equation 2.1b, that is: $z = -ax_1 + \pi x_2$. We seek to drive y (the loop current) to a

target magnitude of 0.15 A. Using estimates $\hat{\alpha}$, $\hat{\gamma}$, the known power supply voltage V, and Equation 2.3, we estimate $\hat{\kappa} = \frac{\hat{\alpha}}{\hat{\gamma}V} = 2.93$ and set a target output value of $Z = \hat{\kappa}\pi(0.15) = 1.38$ when solving CP1.

After solving CP1, arrays $\boldsymbol{m}$, $\boldsymbol{d}_1$ and $\boldsymbol{d}_2$ encoding the multi-mode PWM decomposition of the time-optimal control signal were calculated, and loaded into the memory of a Texas Instruments F28379D microcontroller, which was dedicated to the task of manipulating the H-bridge by running Algorithm 4. Algorithm 4 was executed to excite the oscillator, and measurements of w_1/C and w_2 were taken with an oscilloscope. Plots of the measurement data are shown in blue in Figure 4.3. For comparison, a simulation model of the experimental plant was constructed using the estimated circuit component values of Section 4.2. The continuous-time time-optimal control was computed using the methods presented earlier, and a system response simulation was performed. The results of this system response simulation are overlaid on measurement data, in red, in Figure 4.3. A black dashed line indicates the time instant where the target current must be reached, and a red dashed line indicates the target current level of 0.15 A. Measurement data and simulation data are in agreement, verifying the success of the proposed method.

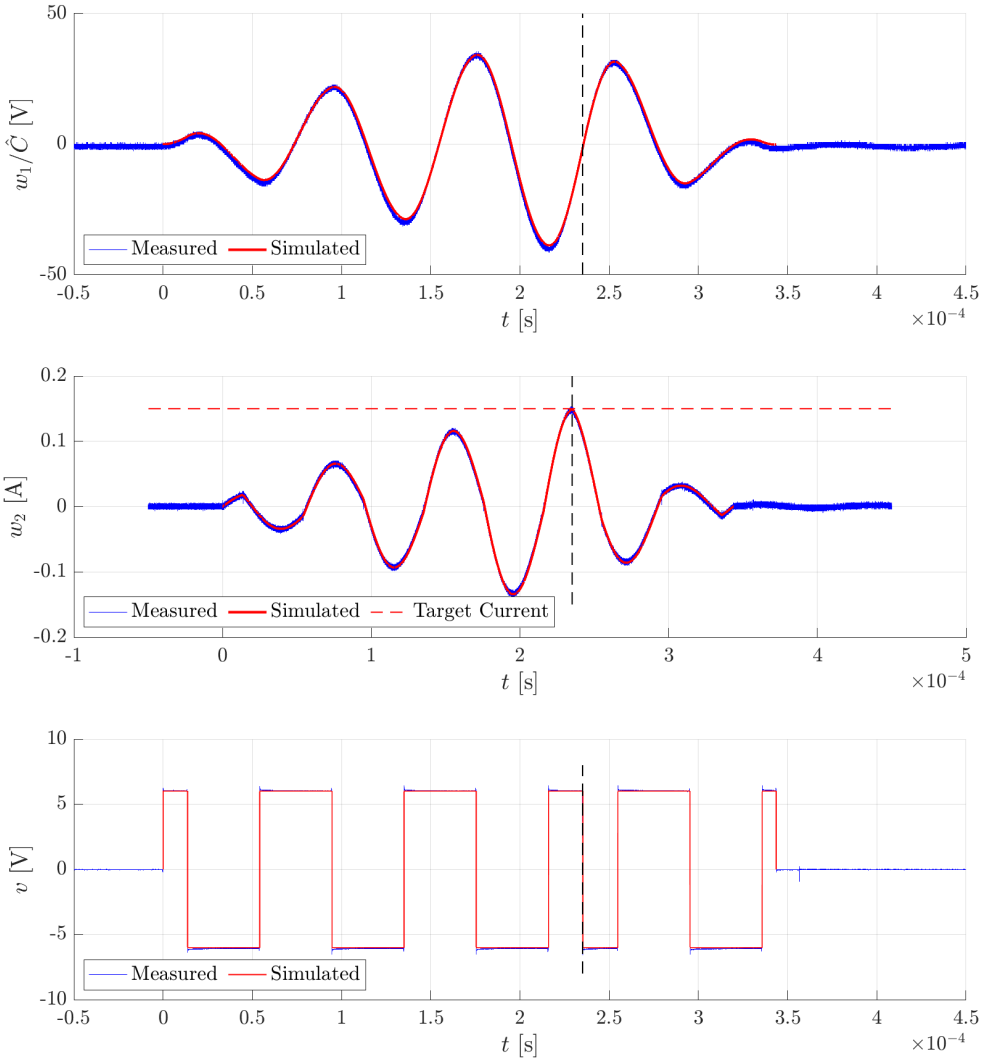


Figure 4.3: Experimental results with simulation results overlaid

CHAPTER 5
SUMMARY AND CONCLUSIONS

Two minimum-time control problems, called Control Problems 1 and 2, were considered in the context of the linear, time-invariant, stable, second-order oscillator. The benefits and applicability of these minimum-time problems to ultrasonics and haptics were highlighted. A simple search algorithm was presented to obtain the solution to either minimum-time problem, and Control Problem 1 was shown to be a preferable formulation from a practical engineering perspective. The use of simple search algorithms to solve time-optimal control problems for second-order oscillators does not appear to be in the existing literature.

A novel method to realize the solution, called *multi-mode PWM*, was presented in detail in Chapter 3. This chapter described how the pulse-width modulation hardware, which is found commonly in modern microcontrollers, can be strategically operated to generate bang-bang control signals. Conventionally, PWM hardware is commanded to generate a bipolar square wave of a set frequency. The hardware allows for the duty cycle of the PWM waveform to be programmed at each switching period, and this feature is commonly used in DC and AC motor control applications. However, existing PWM hardware has additional capabilities that allow for the generation of more exotic wave shapes, and these capabilities are fully exploited in Chapter 3. Ten wavelet-like waveforms, termed "mode waveforms", were defined and used to synthesize control waveforms which were obtained by solving time-optimal control problems for second-order oscillators. Conditions were stated under which the solution to any time-optimal control problem for second order oscillators could be realized using the multi-mode PWM methodology. Based on a review of the literature, the multi-mode PWM appears to be the first result to demonstrate that PWM hardware

commonly found in modern microcontrollers is capable of generating time-optimal control signals.

Bench-top experiments were performed to demonstrate the efficacy of the proposed control method on a physical system, and results were presented in Chapter 4. A voltage-driven series RLC circuit was chosen as the second-order oscillator for demonstration purposes because the state variables of this system can be easily measured using an oscilloscope. Least-squares parameter identification was performed on the physical plant to obtain model parameter values and Control Problem 1 was solved for the bench-top system, with the target set specified in terms of loop current. Experimental results demonstrated the faithful realization of bang-bang control signals using an H-bridge and the satisfactory performance of open-loop control to achieve the desired system response.

Appendices

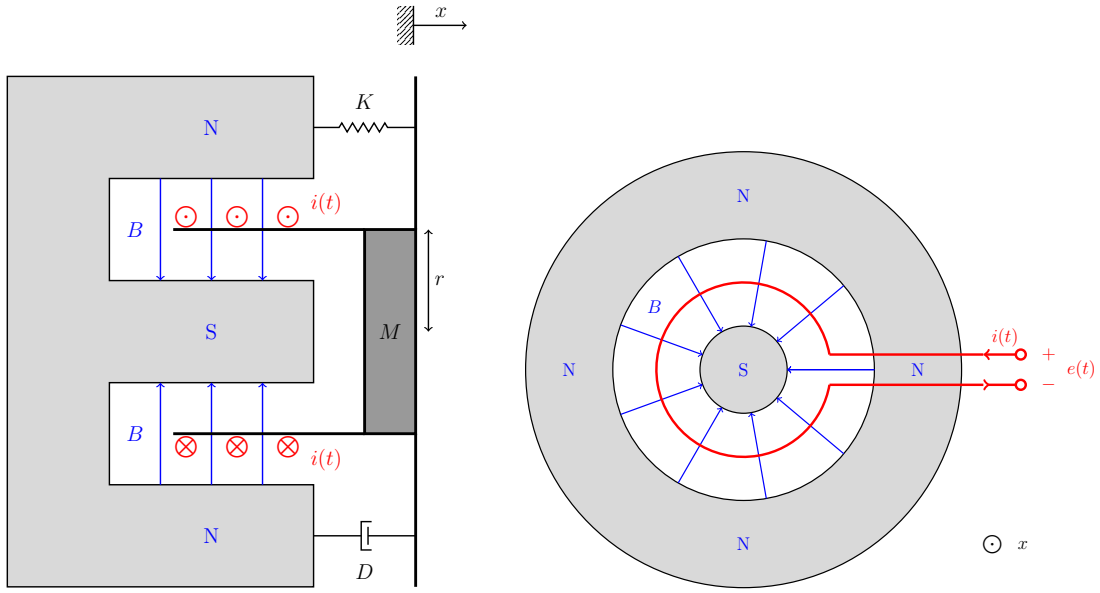
APPENDIX A
SECOND-ORDER MODELS OF SELECT TRANSDUCER SYSTEMS

Second-order dynamic models for some simple electromechanical transducers are developed in this appendix. More complex systems can also be modeled as second-order oscillators under certain conditions. For example, a second-order model for the mechanical dynamics of a capacitive micromachined ultrasonic transducer is given in [16]. Second-order models for a comb-drive microresonator, a vibration sensor, and a force sensor are constructed in [17].

A.1 A Magnetic Field Electromechanical Transducer

A simple magnetic field electromechanical transducer is shown in Figure A.1 [14]. The schematic shown is for a loudspeaker; the permanent magnet is stationary. The linear resonant actuator used in haptics is closely related. In this case, the coil is fixed and the permanent magnet is in motion.

A moving mass M of radius r is rigidly attached to a coil consisting of N turns of wire. The mass is attached to a stationary permanent magnet via a spring with spring constant K and a viscous damper with damping coefficient D. The coil is modeled by a lumped resistance R and lumped inductance L. The two ends of the coiled wire are driven by a voltage $e(t)$. Interactions between the current-carrying wire and the magnetic field B due to the permanent magnet result in a motive force being applied to mass M. The following second-order state-space model is adequate for this system if dynamics associated with coil inductance can be neglected due to time constant



(a) Side view (b) Front view

Figure A.1: Schematic of a magnetic field electromechanical transducer

separation between the electrical and mechanical subsystems (i.e. $L/R \ll M/D$):

$$\begin{bmatrix} \dot{x}(t) \\ \ddot{x}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{K}{M} & -\frac{DR+C^2}{MR} \end{bmatrix} \begin{bmatrix} x(t) \\ \dot{x}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{C}{MR} \end{bmatrix} e(t)$$

where $C = B(2\pi r N)$, and dots above a variable represent differentiation with respect to time.

A.2 An Electric Field Electromechanical Transducer

The electric field dual of the magnetic field transducer of Figure A.1 is shown in Figure A.2 [15].

A moving, conducting plate of mass M is attached to a stationary conducting plate via a spring with spring constant K and natural length ℓ, as well as a viscous damper with damping coefficient D. Both conducting plates have area A, and are separated by an air-filled region with permittivity ε_0. A voltage difference $v(t)$ is imposed between the two plates. A motive force is applied to the moving plate due to

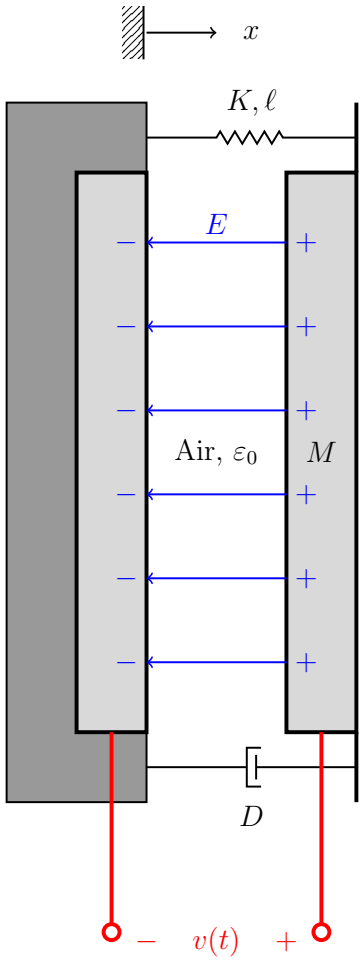


Figure A.2: Schematic of an electric field electromechanical transducer

the interaction between the charge present on the moving plate, and the electric field that is produced by the charge on the stationary plate. If the system is operating about an equilibrium point ($x = X, v = V$) - i.e.:

$$x(t) = X + \tilde{x}(t) \quad , \quad \dot{x}(t) = 0 + \dot{\tilde{x}}(t) \quad , \quad v(t) = V + \tilde{v}(t)$$

then the following linearized second-order state-space model is adequate:

$$\begin{bmatrix} \dot{\tilde{x}}(t) \\ \ddot{\tilde{x}}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{K_o}{M} & -\frac{D}{M} \end{bmatrix} \begin{bmatrix} \tilde{x}(t) \\ \dot{\tilde{x}}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{C_o E_o}{M} \end{bmatrix} \tilde{v}(t)$$

where $C_o = \frac{\varepsilon_0 A}{X}$, $E_o = \frac{V}{X}$, $K_o = K - C_o {E_o}^2$, and dots above a variable represent differentiation with respect to time.

A.3 A Piezoelectric Ultrasonic Transducer

Piezoelectric ultrasonic transducers are resonant systems which convert energy between the electrical, mechanical, and acoustic domains. In a region of frequencies around resonance, ultrasonic transducers are well modeled by the third-order Butterworth Van-Dyke model. This model is in the form of an electric circuit, and contains two parallel branches. Figure A.3 shows this equivalent circuit with an additional series resistance R_0:

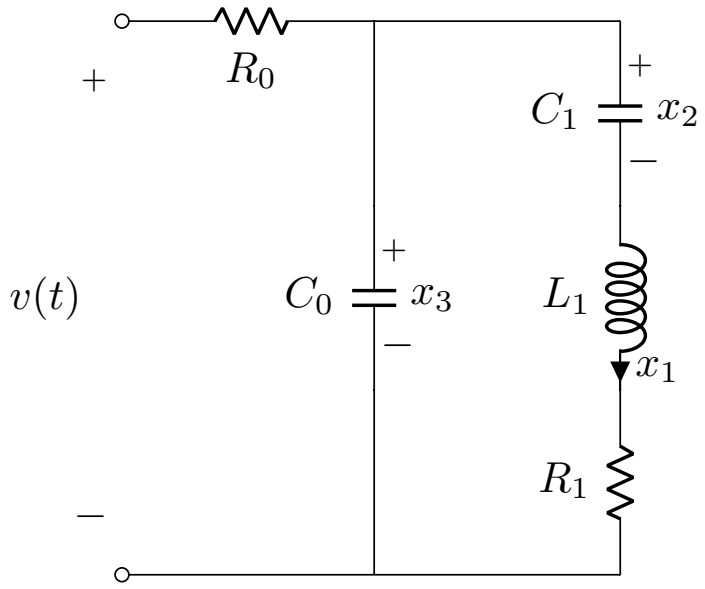


Figure A.3: Equivalent circuit model of a piezoelectric ultrasonic transducer

The first parallel branch, containing capacitor C_0, represents a physical capacitance associated with a slab of piezoelectric material inside the transducer. The mechanical subsystem of the transducer is modeled as a second-order mass-spring-damper system. In the second parallel branch of Figure A.3, C_1 and L_1 represent the mass-spring ensemble, and R_1 represents the damper, by means of analogy.

Application of Kirchoff's laws to the circuit in Figure A.3 yields the third-order

state space model:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -\frac{R_1}{L_1} & -\frac{1}{L_1} & \frac{1}{L_1} \\ \frac{1}{C_1} & 0 & 0 \\ -\frac{1}{C_0} & 0 & -\frac{1}{R_0 C_0} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{R_0 C_0} \end{bmatrix} v$$

where dots above a variable represent differentiation with respect to time. If R_0 is sufficiently small to ensure well separated time constants $R_0 C_0 \ll L_1/R_1$, then we may make a quasi-steady-state assumption: $\dot{x}_3 \approx 0$. This in turn implies:

$$\begin{aligned} \implies & x_3 \approx v - R_0 x_1 \\ \implies & \dot{x}_1 = -\tfrac{R_1}{L_1} x_1 - \tfrac{1}{L_1} x_2 + \tfrac{1}{L_1} x_3 \\ & = -\tfrac{R_1 + R_0}{L_1} x_1 - \tfrac{1}{L_1} x_2 + \tfrac{1}{L_1} v \end{aligned}$$

The resulting reduced-order state-space model is then:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -\frac{R_1 + R_0}{L_1} & -\frac{1}{L_1} \\ \frac{1}{C_1} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{L_1} \\ 0 \end{bmatrix} v$$

Note that this corresponds to a series RLC circuit driven by a voltage source, where x_1 is the loop current, and x_2 is the capacitor voltage.

REFERENCES

[1] M. Athans and P. L. Falb, *Optimal Control: An Introduction to the Theory and Its Applications*. McGraw-Hill, 1966.

[2] M. Namavar, A. J. Fleming, M. Aleyaasin, K. Nakkeeran, and S. S. Aphale, "An analytical approach to integral resonant control of second-order systems," *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 2, pp. 651–659, 2014.

[3] S. S. Aphale, A. J. Fleming, and S. O. R. Moheimani, "A second-order controller for resonance damping and tracking control of nanopositioning systems," in *Proceedings of the International Conference on Adaptive Structures and Technologies*, Ascona, Switzerland, 2008.

[4] R. J. Moon, A. San-Millan, M. Aleyaasin, V. Feliu, and S. S. Aphale, "Selection of positive position feedback controllers for damping and precision positioning applications," in *Asian Simulation Conference*, Melaka, Malaysia, 2017, pp. 289–301.

[5] N. C. Singer and W. P. Seering, "Preshaping command inputs to reduce system vibration," *Journal of Dynamic Systems, Measurement, and Control*, vol. 112, no. 1, pp. 76–82, 1990.

[6] W. E. Singhose, W. Seering, and M. Singer, "Input shaping for vibration reduction with specified insensitivity to modeling errors," *Proc. Japan-USA Symp. Flexible Automation*, vol. 1, 1996.

[7] T. Tuttle and W. P. Seering, "A zero-placement technique for designing shaped inputs to suppress multiple-mode vibration," *Proceedings of 1994 American Control Conference*, vol. 3, pp. 2533–2537, 1994.

[8] K. L. Sorensen, K. Hekman, and W. E. Singhose, "Finite-state input shaping," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 3, pp. 664–672, 2010.

[9] M. Buhl and B. Lohmann, "Control with exponentially decaying Lyapunov functions and its use for systems with input saturation," in *2009 European Control Conference*, 2009, pp. 3148–3153.

[10] R. A. Braker and L. Y. Pao, "Proximate time-optimal control of a harmonic oscillator," *IEEE Transactions on Automatic Control*, vol. 63, no. 6, pp. 1676–1691, 2018.

[11] *DRV2603 Haptic Drive With Auto-Resonance Detection for Linear Resonance Actuators (LRA) datasheet*, Dallas, Texas: Texas Instruments, 2016.

[12] *DRV2603 ERM/LRA Haptic Driver Evaluation Kit user guide*, Dallas, Texas: Texas Instruments, 2013.

[13] L. Y. Pao and W. E. Singhose, "Robust minimum time control of flexible structures," *Automatica*, vol. 34, no. 2, pp. 229–236, 1998.

[14] E. Cheever, *Lecture notes for linear physical systems analysis*, 2019.

[15] H. Woodson and J. Melcher, *Electromechanical Dynamics*, ser. Electromechanical Dynamics pt. 1. Wiley, 1968.

[16] K. Brenner, A. Ergun, K. Firouzi, M. Rasmussen, Q. Stedman, and B. Khuri–Yakub, "Advances in capacitive micromachined ultrasonic transducers," *Micromachines*, vol. 10, no. 2, pp. 152–179, 2019.

[17] H. A. C. Tilmans, "Equivalent circuit representation of electromechanical transducers: I. lumped-parameter systems," *Journal of Micromechanics and Microengineering*, vol. 6, no. 3, 157–176, 1996.